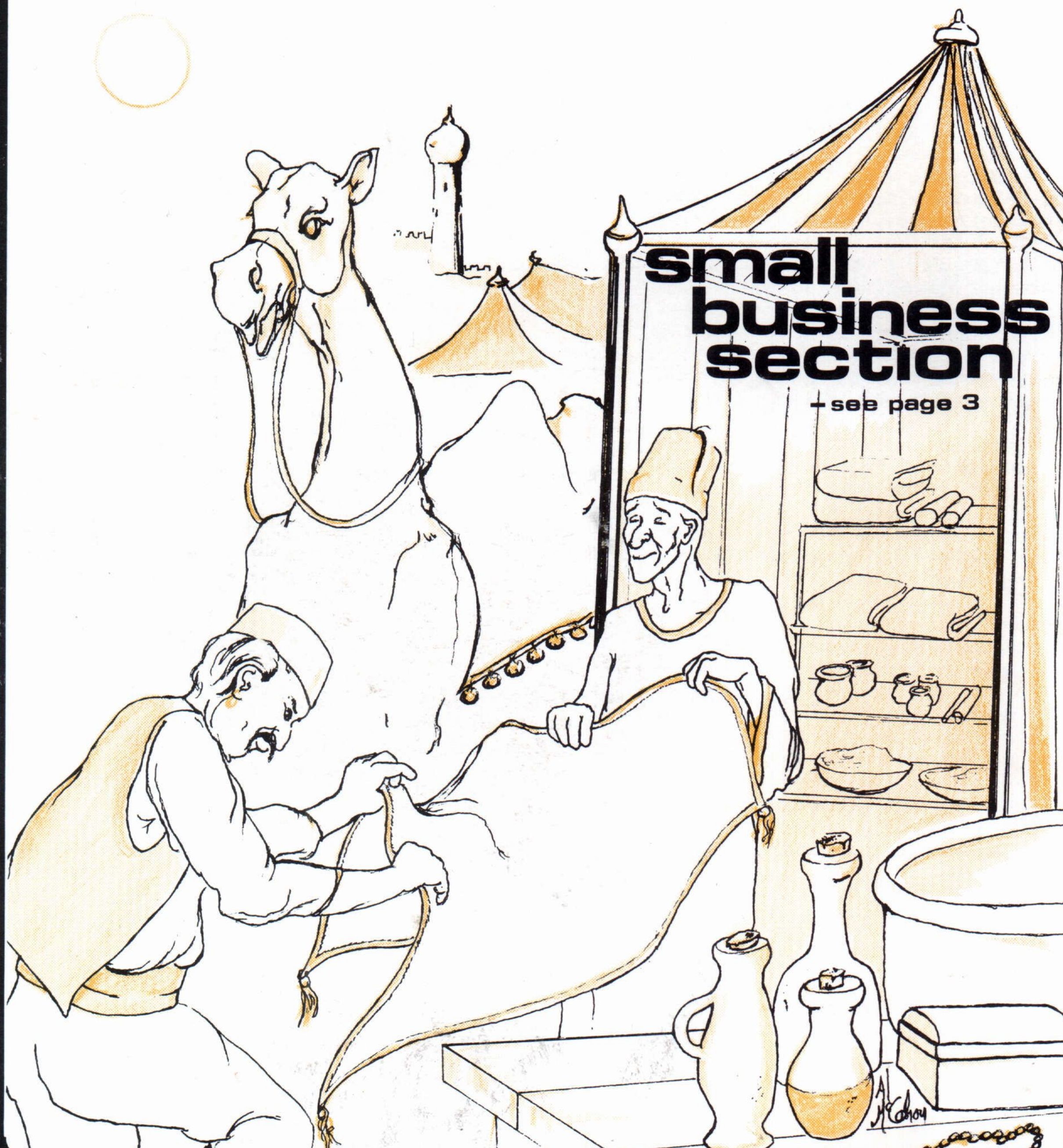
computer notes

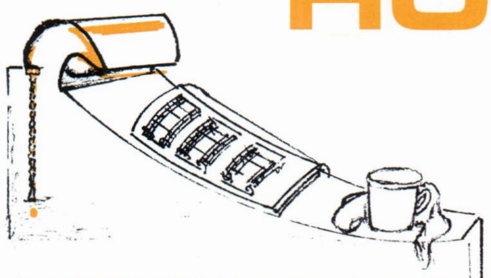
May '77

Volume 2, Issue 11

50¢



BLOW YOUR OWN HORN



COMPUTER NOTES is continually seeking quality manuscripts on applications, trouble-shooting, interfacing, software, book reviews, fiction, cartoons and a variety of other computer-related topics.

Articles should be a minimum of 800 words and a maximum of 3600 words long (about 15 pages typed double-space). Honorariums are based on an article's technical quality and its suitability for *C.N.'s* readership. Payment will range from 50¢ to $1 per typeset magazine column inch for all text and programs. No payment will be made for illustrations. All articles are subject to editing to fit space requirements and content needs of our readership. Payment for articles which are accepted will be sent upon publication.

Articles submitted to *C.N.* should be typed, double-space, with the author's name, address and the date in the upper left-hand corner of each numbered page. Authors should also include a one-sentence autobiographical statement about their job, professional title, previous electronic and/or computer experience under the article's title. Authors should retain a copy of each article submitted.

Photos, charts, programs and figures should be clearly labelled and referred to by number within the text of the manuscript. Only clear, glossy black and white photos (no Polaroid pictures) can be accepted. All photos should be taken with uniform lighting and sharp focus.

Program listings should be recorded with the darkest ribbon possible on blank white paper.

Computer Notes
Attn. Linda Blocki
2450 Alamo SE
Albuquerque, New Mexico 87106

Articles which are not accepted will be returned.

COMPUTER NOTES is published monthly by MITS, Inc., 2450 Alamo SE, Albuquerque, NM, 87106, (505) 243-7821. A free year's subscription is included with every purchase of an Altair™ computer. Regular subscriptions can be ordered from the MITS Customer Service Dept. for $5 per year in the U.S. and $20 per year for overseas. Single copies are available for 50¢ each at all Altair Computer Centers. Entire contents copyright, 1977, MITS, Inc. Send articles, questions, comments and suggestions to Editor, *COMPUTER NOTES*, MITS, Inc.

BITS AND PIECES

By Sondra Pollini

I'd like to remind readers that the purpose of this column is to explain MITS marketing policies and procedures, so we welcome your feedback. Send questions and suggestions to Sondra Pollini, Marketing Department, MITS.

Time Payments

We ship Time Payment segments only when previously notified by customers. Without notification, no shipments will be made. If you are unable to order a segment or wish to order more than one per month, please notify us. With any request for shipment of another segment, be sure to include both the name under which your original order was placed and the MITS order number.

Software

Most software questions can be answered more quickly and efficiently if customers contacting our Software Department will provide the following information.

1. The Grade of BASIC being used. (4K, 8K, Ext., or Disk)
2. The Version of BASIC being used. (3.3, 3.4, 4.0)
3. Where you purchased BASIC.
4. A complete explanation of the problem(s).
5. A computer listing of the errors that occur when you run BASIC or utilize a BASIC program.

Applications Requested

With the use of Altair microcomputers increasing rapidly, we would like more information on our customers' applications. All interested customers should submit a brief explanation of their Altair system configuration and application to the MITS Marketing Dept., c/o Sondra Pollini.

Troubleshoot and Modify Your Altair 88-ACR

By Rich Haber

Third in a three-part series on the Altair 8800 ACR.

The first two parts of this series covered the theory and operation of the Altair 8800 ACR module. In this part we will examine troubleshooting procedures and suggest some modifications for the SIO section of the Altair 8800 ACR.

If you are having trouble with the 88-ACR, try to isolate the problem with the following checks:

SIOB Troubleshooting

Routine checks:

1. Make sure that all connections are congruent with the strapping diagrams (see Figures 2 and 3).
2. Is the board a Rev. 1? If not, modify it according to the directions given in the section on SIOC Rev. 0 MOD.
3. Check to see if R4 is a 7.5K resistor.
4. Make sure that the interrupt jumper is not installed.
5. Check +5 and -12 supplies.
6. Check the clock input for the waveform at M-40.

If you cannot input data, deposit the bootstrap and play 125's from the test side of the tape.

1. Do you get an input at pin 20 of UART?
2. Is D8 pulsing low to enable data in tri-state buffers?
3. Is pin 4 of UART ($\overline{\text{RDS}}$) pulsing low to enable data transfer?
4. Is pin 18 pulsing low to enable status transfer to the CPU?
5. Is pin 19 (DA) pulsing high to tell the CPU that data is available?
6. Are pins 15, 14 and 13 (OVR, FE, PE) staying low to indicate that there are no transmission errors?

If just one bit is not being transmitted, you may not notice it playing 125's. Try playing 256's. If you see a bit missing on the data window, check the corresponding buffer and gate.

If you cannot transmit data, load the output from the sense switches program listed earlier. Single step through the program. Check for the following:

1. To output pin 23, ($\overline{\text{TDS}}$) must go low
2. Pin 22 (TBMT) should go high after an output sequence

Continued

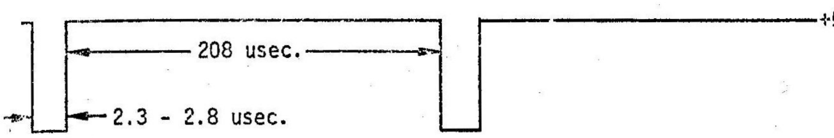


Figure 1

mits
2450 Alamo S.E.
Albuquerque, New Mexico 87106

IN THIS ISSUE

3. When the program is run, continuous data-out should appear at pin 25 (TSO)

If proper command signals aren't getting to the UART, trace back through the logic gates to find the cause.

There is an esoteric "bug" in the timing. An instruction address after an input instruction pair must not be the address of any I/O port being used in a system. The following program cannot occur:

Address	Data Byte
000	xxx (don't cares)
001	xxx
002	xxx
003	xxx
004	xxx
005	333
006	006 or 007
007	xxx

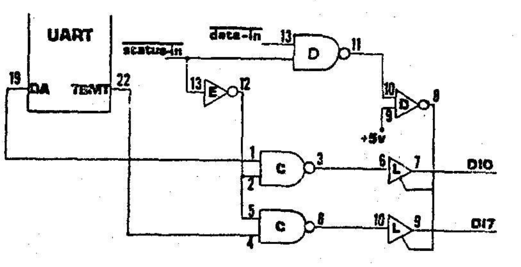
When the CPU is trying to input data and the address changes to 007, it will reset the UART in the middle of a byte and cause that byte to get lost. (Note: This only occurs when you use your own programs for the ACR.)

SIOB Board Modifications:

First, check your board to see that pin 6 of IC O is connected to pin 11 of ICs P, Q and R. If it is not, install jumpers.

SIOC Modifications:

The second modification of the Rev 0 board (called hardware interrupt) is in the status provided to the data bus by the UART. The new board will pull DI0 low on the DA high condition (input status) and will pull DI7 low on favorable output status. The logic follows.



The leads to pins 19 and 22 of the UART must be cut with an EXACTO knife or razor blade. (This necessitates removing the UART socket, which is a difficult procedure for a novice. So it's better to send the board to a qualified technician.) Then install jumpers between pin 19 and C1 and pin 22 and C4.

Modification of the 88-ACR Modem Board:

The changes described below will allow the 88-ACR to accept 2.75 times wider speed variation when demodulating tapes written with the new method. However, demodulation (reading) of tapes written by the old method will be the same.

Purpose: Make reading and writing of data on audio tapes less susceptible to errors due to speed variations and to make adjustment of R29 (phase locked loop center frequency adjust) less critical.

Method: Change modulator frequencies from 2225Hz/2025Hz-(200 Hz difference) to 2400Hz/1850Hz-(550 Hz difference). This change keeps the center frequency at 2125Hz, allowing the 88-ACR to demodulate (read) either type of modulation.

Modifications to 88-ACR Modem Boards:

A) Modulator: Change jumpers as follows:

1. Remove jumpers #1 & 2.
2. Connect pins 3, 4 and 5 of IC "J" together.
3. Change jumper #3 from 3B to 2A.
4. Change jumper #4 from 4B to 4A.
5. Disconnect pins 5 and 6 of IC "K" from ground (unsolder and bend out of board).
6. Connect pins 4 and 5 of IC "K" together.
7. Change jumper #5 from 5B to 2A.
8. Connect pin 6 of IC "K" to point 5A.
9. Change jumper #7 from 7B to 7A.

NOTE: The "B" row of jumper points is closest to the edge of the Modem Board. The "A" row of jumper points is closest to the row of numbered jumper wires (see schematic diagram in manual).

This changes the modulation frequencies to:

LOGIC 1 = 2404 Hz $\pm$ 1 Hz
LOGIC Ø = 1852 Hz $\pm$ 1 Hz

(measured at IC "H"-8)

B) Demodulator: Change R28 to 3.3K ohms, or parallel a 5.6K ohm resistor with the existing 8.2K ohm resistor.

This change increases the lock range of the phase locked loop (IC "C") for the wider frequency spread of the new modulation method. It does not affect demodulation of tapes previously recorded with the old frequencies (2225/2025 Hz).

This change allows tape speed variations between writing and reading of over 3%, without readjustment of R29 (if demodulating tapes written with the new method).

Other Circuitry Changes Recommended for the 88-ACR

A) Change C18 (was 5 µf electrolytic) to a 1 µf mylar or non-polarity sensitive character. This prevents breakdown of C18 when reverse biased (no carrier).

B) Use the old C18 (5 µf electrolytic) to add a 5 µf capacitor: + end to IC "C" pin 9 end of R30, -- end to -12 volts. This helps stabilize adjustment of R29.

C) Change R32 to 8.2K (use old R38) and change Z1 (12 volt zener) to a 3.3K resistor. This allows the P. L. L. output (IC "C", pin 8) to pull down point "RS" to a valid logic Ø even if the system negative voltage supply is low.

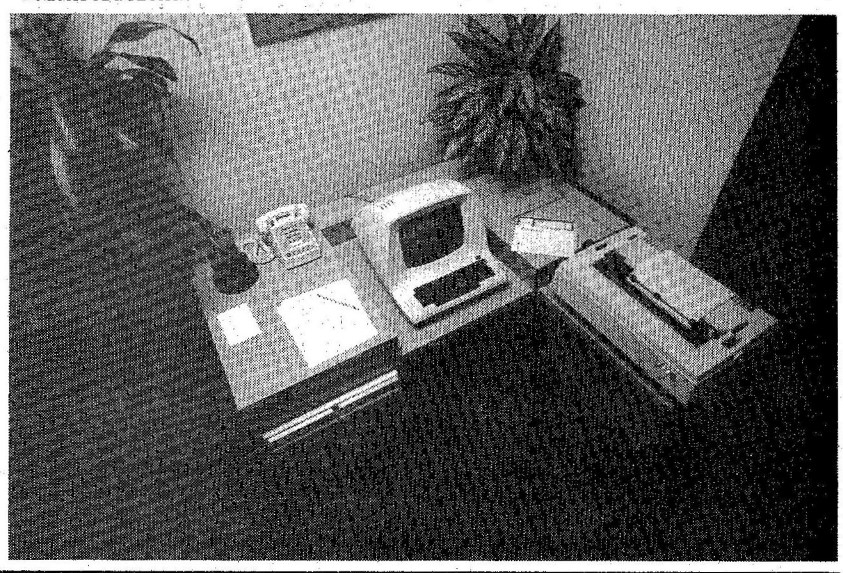
D) Remove diode D4. This allows reading and writing of tapes simultaneously.

E) Optional - For indication of the carrier (2K Hz tones) a L. E. D. may be wired to points "A" and "K" on the Modem board. Remove the jumper wire from "A" to "K" and connect the LED anode to "A," the cathode to "K." When the carrier is being received, the LED forward current is about 10MA. Use a red LED only--1.7 volts forward drop.

All 88-ACR kits shipped after March 15, 1976 contain the modification described above.

small business section

In the November issue of COMPUTER NOTES, MITS announced the formation of the Altair Software Distribution Company and introduced the new business system software to be distributed through the Altair Computer Center network. This issue contains several articles that describe the business software packages in greater detail. To find out more about each package, stop by your nearest Altair Computer Center and ask for a demonstration.



Altair Computers Assist Environmental and Power Companies

By Jeanne Pease

Managing Editor
COMPUTER RETAILING

Two companies in Georgia recently developed unique energy and environmental applications for their Altair computers. Georgia Power Company, a division of the Southern Company -- a utility holding company, is using a number of Altair microcomputers to assist with its Power Management System (PMS II). Municipal & Industrial Pipe Services, Ltd. (MIPS) is conducting water flow analysis with an Altair microcomputer.

MIPS is an environmental service company located in Douglasville, Georgia, a suburb of Atlanta. MIPS uses an Altair 8800a to analyze data gathered from their Environmental Monitoring System, a network of monitoring stations used by municipal and industrial water and sewer service producers.

The principle application of the system is to produce a histogram which shows the current water use, waste return and rainfall. Sections of the graph may be enlarged. Multiple graph capability is also used for comparisons between histograms. All output is initially displayed on their Hazeltine CRT and may then be diverted to the Qume printer if a hard copy is desired.

The second application is the transmission of data from the Environmental Monitoring System to a time-sharing system for data reduction. Special hardware and custom software is required.

In addition to these two applications, MIPS uses their system for accounting, inventory and word processing.

Continued

Altair Word Processing Package Offers Special Formatter Commands

By John Hayes
General Manager
ASDC

Word processing systems come in all shapes, sizes, capabilities and prices and are generally used for the storage and repetitive typing of business documents. But word processing systems differ in the way they implement this basic function and in the additional capabilities they offer.

Even the most simple systems provide for some editing of text, usually as data is being entered and at some later time. However, users are usually more interested in a system's ability to make more extensive editing changes--moving portions of text, inserting new text, deleting old text, providing access to text that is stored in other documents and being able to search text for a particular character set.

The Altair Business System Word Processing Package contains a number of features not found in other word processing systems. The special formatter commands take the package from the arena of a simple office text editing system into the realm of a sophisticated, high level computer language for controlling text. Special routines allow the Word Processing Package to accept virtually any file of text, including BASIC programs in ASCII format, as input. These and other features enable the system to far surpass what is required of a simple office system.

The Word Processing Package is a flexible and powerful text editing and formatting system. It operates on an Altair 8800 computer equipped with 48K of memory, a processing printer, a video terminal and at least one floppy disk drive. (Most users find a dual drive installation more effective.) Since the package utilizes Altair Extended Disk BASIC plus several machine language routines, it can be used for a wide variety of other purposes, including the operation of the Altair Business System Accounting System Packages described elsewhere in this issue of COMPUTER NOTES.

Continued

Altair Word Processing Package Offers Special Formatter Commands

Continued

At first glance, the list of commands for the Package's editor or formatter may appear somewhat imposing. But the package is self-instructing, so even an inexperienced user can effectively use it in just a short time.

The remainder of this article will cover a number of features of the Word Processing Package. But space does not permit an explanation of all the powerful commands. Your local Altair Computer Center can provide more complete information.

The Word Processing Package consists of two modules--the text editor and the text formatter. The editor allows the user to create, modify and store the document. A set of editor commands allows the user to control the operation of the system as text is being input or modified at a later time. A set of formatter commands directs the way in which the formatter prints the document. The formatter commands are visible to the user as a set of special characters and statements. The user can see the embedded commands, not merely their result, in the raw document. This gives the user much more control over the document to be printed.

The editor handles text as lines of up to 120 characters. Each line of text has a line number in the range .001 to 999.999. The maximum number of lines permitted in a single document is 1,000. So, the size of a document can be up to 120,000 characters or about 40-45 single-spaced, letter-sized pages. Several documents may be handled together to form a "book" of virtually unlimited length. The Word Processing Package uses fractional line numbers to allow text to be inserted at a particular point without altering subsequent line numbers. Editing may be performed on a line of text as it is entered or at some later time with in-line editing done by standard cursor control keys, if the video terminal is so equipped.

The Word Processing Package provides a variety of other editing commands which facilitate the creation and maintenance of text.

The editor operates in two modes: (1) the text mode--entry of text lines and (2) the edit mode--insertion of an Editor Command. The editor flip-flops between the two modes when a null string, a blank line, is typed.

When the editor is expecting a line of text, the next sequential line number is presented by the system. A "?" is displayed as a prompting character when the editor is expecting an EDIT command.

The user initiates the editing of a document by identifying the document name (and the disk drive number if more than one disk drive is employed) with an EDIT (file name) command. If a document by that name already exists on the diskette mounted in the specified drive, that document file is "opened" for editing. If not, the system opens a new document file under that name.

When a document is open for editing, a number of commands are available to find and manipulate the text. If a command is typed incorrectly, the system will respond with a brief prompt of the correct form of the command. At any time a user can type the HELP command to receive more detailed information about the error or other features of the system.

The FIND command allows all or part of the document to be searched for a particular set of characters or words, called a string. The system can be instructed to find and print all occurrences of the string or only the first occurrence. The string can be defined by using single or double quotes. This will locate the exact character sequence either isolated from other characters by delimiters, such as spaces, or wherever found in the text. For example, <u>FIND ALL "jack" 1-100</u> would find all occurrences of the word jack in lines 1 through 100 only if jack were preceded and followed by a delimiter, such as a space or a period. This command would not find the work jack in jacket. However, <u>FIND "jack'</u> would find jacket but not blackjack, and <u>FIND 'jack'</u> would find jack in both jacket and blackjack.

Two other commands that are frequently used are the COPY command and the MOVE command. Both commands copy a specified set of lines to another location with a specified line number increment. The MOVE command deletes the old lines. The LIST command displays, on the video terminal or printer, all or part of the document exactly as the document is stored in the system--complete with line numbers and embedded format command. The LIST command should be compared to the PRINT command, which causes the embedded formatter commands to be acted upon. LIST is used to review and edit a document.

The power of the Word Processing Package is greatly expanded through the use of star files. A star file is a document that is called upon as a reference or source document for the document currently being edited. Text in the star file can be listed (with the LIST* command), searched (with the FIND* command) and copies to the file being edited (with the COPY* command). However, the system will not permit any text in

Altair Computers Assist Environmental and Power Companies

Continued

Georgia Power Company's PMS II project involves the study of the future reliability of the Southern Company's Power Management System, a computer-oriented communications network for transmission, maintenance and operation information. Two Altair 8800a's and four Altair 8800b computers are being used in the project. They are interfaced with Altair 4-port parallel boards, Processor Technology VDM-1 boards and various mini floppies. PMS II will be implemented this August.

Under the present PMSI System, an operator in each division of the power transmission network provides the PMS with generating system information. All supervisory control and data acquisition (SCADA) functions are performed by a large IBM computer in Southern's Birmingham headquarters. Any operator needing information about his or another transmission system can request a circuit printout on a CRT terminal. But even if the system is in his own backyard, the information must travel from the substation, through the computer in Birmingham and then back to the division operator.

Research indicates that usage of the Birmingham system is increasing rapidly. Olin Williams Jr., Georgia Power Computer Control Engineer and head of the PMS II project, said that PMS II will relieve part of the load on the central computer. He said this can be accomplished by distributing the processing of division SCADA through multiple microcomputers. This will provide each division with an independent information processing system.

A computer cluster located inside each division will intercept any signal, giving the operator direct access to all substation information. Since all data is currently transmitted to Birmingham via microwave link, the PMS II system will eliminate the problem of unavailable information due to wind-damaged or otherwise inoperative microwave towers.

So the Altair microcomputers will decrease communication costs and improve SCADA reliability by eliminating potential demand overload.

the star file to be modified so that a master document can be kept secure.

During the editing of another document, any number of documents can be opened, one at a time, as star files. But both files must be on diskettes that are loaded in disk drives. (Hence, the desirability of a two-disk drive system.)

The editor has a number of other useful commands to facilitate text editing. For example, lines may be inserted, deleted or renumbered at any point. The command statement REPLACE ALL "stock#" WITH "Harris Supply Inventory Control Number" would make the indicated replacement throughout the document wherever "stock#" occurred. Editor commands may be abbreviated to simplify typing.

Another useful command is the SUBSTITUTE command, which allows text strings to be represented with 10 substitute codes (the % sign followed by a digit 0 to 9). The operator can then use the replacement code instead of typing the whole string.

The formatter is envoked when a PRINT command is issued in the editor. The formatter processes any specified parts of the document and types out the document under the control of the embedded formatter commands.

When the text is processed by the formatter, the margins are set, text is "filled" by flowing from one line to the next, pages are automatically spaced and numbered and specified headings are typed on each page. A new page can also be "forced" to start whenever a user desires.

The formatter commands are special strings which are input and modified in the editor. It is not necessary to begin a new line every time a formatter command is used. The commands may appear anywhere in the text. Commands may be "nexted" so that several particular formats will apply, such as boldface and underlining, to a text string.

When a user first initializes the Word Processing Package, a set of default parameters are defined for page size, margins and other formatting information. These values must be redefined only if values in a document differ from the standard.

The formatter contains all the usual formatting commands, such as underlining, centering, indentation, headings, pagination at a specified place, right justification, paragraph and page definition, skipping lines, multiple spacing and tabulations. A number of special formatting commands are also available. They justify text in various ways and define special emphasis type styles.

A specified number of lines can be "floated." If there is no space on the existing page, space will be reserved on the next. This feature is particularly useful for attaching photos or drawings to the document.

The real power of the Word Processing Package comes not from these standard formatting commands but from the file and input controls contained in the formatter.

The first group of these formatter commands allows user interaction with the document while it is being printed. The @DISPLAY command allows a specified string of text to be displayed on the video terminal but not printed in the document. User defined prompts displayed on the terminal can be very helpful. Another command allows interactive manipulation of text as it is being processed.

The second major group of special formatter commands involves the use of a star file in printing. Just as a user may have star files opened to provide text for a document being edited, the document may access other documents one at a time for text and other commands. By using these features, a user may build "procedure" or "control command" files.

The third major set of special formatter commands is the set of text variables. These variables represent other strings of text. The system recognizes 10 variables. The text strings that each of these variables represents may be specified within the document by a special command or may be specified in the editor by the ?PRINT command initiated during document printing. The text string represented by a variable may be text or may contain formatter commands and variables.

Altair Payroll Package Supports 500 Employees on One Floppy Disk

By John Hayes
General Manager
ASDC

The Payroll Package is one of four packages which comprise the accounting system portion of the Altair Business System. The Payroll Package will operate in either a stand-alone mode, independent of other packages, or with the General Ledger Package--the heart of the Altair accounting system, for integrated monthly reporting.

Like the other units of the Altair Business System, the Payroll Package requires an Altair 8800 computer with 48K of memory, a video terminal for data entry and review, a line printer or precision printer (preferably with tractor feed) and one floppy disk unit (a dual-disk system is recommended).

The Payroll Package will handle up to 500 employees on a single floppy disk system. Each employee record allows as many as seven deductions to be accumulated from pay period to pay period. Each employee record also contains: (1) the name and address of the employee, (2) state and local codes for tax information, and (3) separate exemption fields for federal and state withholding, marital status, base pay rate and current regular, overtime and other hourly pay fields. Each employee is identified by a six-character, alphanumeric code. The first three characters are the department code, and the latter three identify the employee within that department. Department and employee codes need not be numeric, but each must be unique.

The Payroll Package allows employees to be compensated by hourly, salaried and commission-plus draw methods. These employees may be paid either weekly, bi-weekly, monthly, semi-monthly or by any combination of these periods. Employee records also maintain information on an employee's overtime work. The system provides for the automatic calculation of taxes, deductions and net pay for all employees. It also provides for the individual adjustment of particular employee's taxes, other deductions, regular or overtime pay or any other aspect of that particular employee's pay record prior to the printing of checks.

The system will handle payroll deductions for all 50 states and up to 20 local governments. The tax information is maintained by a set of tax calculation routines for each of the 50 states and 20 local governments. The payroll system provides a complete set of tax tables which enable it to automatically handle employees in many different states. A tax table maintenance routine allows the user to update existing tax tables to accommodate changes in the rates. The tax table files are used by a withholding routine that can calculate any form of payroll withholding tax currently used in the U.S. by federal, state or local governments.

Continued

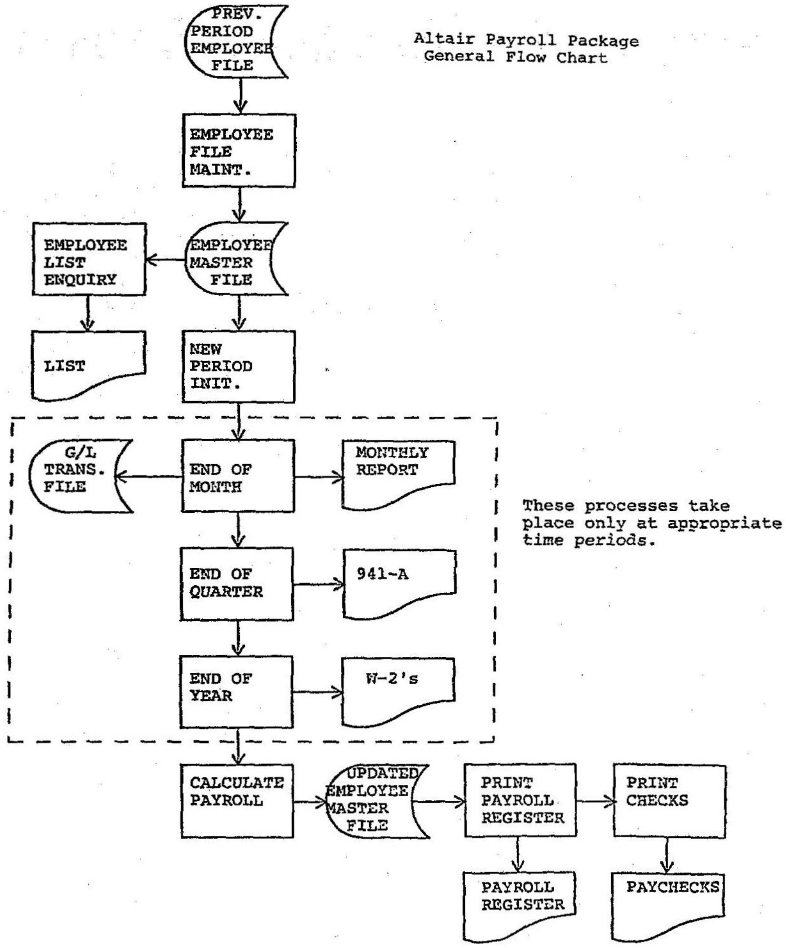
Altair Payroll Package Supports 500 Employees on One Floppy Disk

Continued

Complete current, month-to-date, quarter-to-date and year-to-date information in addition to withholding tax figures and other deduction totals are kept on each employee.

The major reports generated by the system include an employee file listing which provides complete listing to date on a particular employee, a payroll register that provides information for a current payroll, a monthly manager report and the appropriate 941A and W-2 forms. The system automatically types payroll checks or allows a user to manually prepare the checks and then enter information into the system. A standard set of forms is included with the Payroll Package to allow a user to quickly initialize and maintain the payroll system.

When the Payroll Package is used as part of the integrated accounting system, it generates a set of entries which are transferred to the General Ledger Package by account. The system also provides the general ledger with the detailed expenditures for as many as 20 different departments, the net payroll for employees, the net information on FICA, federal and state withholding and other deduction totals. This feature allows the payroll system to automatically provide the journal entries for the general ledger.



Altair Payroll Package General Flow Chart

Additional information about the Payroll Package can be obtained from the local Altair Computer Centers located throughout the United States or from the Altair Software Distribution Company, Suite 343, 3330 Peachtree Road, Atlanta, Georgia 30326.

Altair General Ledger Package Insures Accurate Auditing

By John Hayes

The General Ledger Package is the heart of the Accounting system portion of the Altair Business System. It provides the foundation for a small business financial control system. The entire accounting system consists of the General Ledger Package, a Payroll Package, an Accounts Receivable Package and an Accounts Payable Package.

The General Ledger Package is designed to interface with the other components of the accounting system. It may also be used independently of the other packages in a stand-alone mode.

It provides a small business with the capability of maintaining a detailed general ledger as well as printing a monthly earnings statement (Profit and Loss Statement) and a balance sheet. A Department Profit and Loss Statement is also available. Like the other packages in the Altair Business System, the General Ledger Package requires an Altair 8800 computer with 48K of memory, a video terminal for data entry and review, a line printer or precision printer (preferably with tractor feed) and at least one floppy disk unit. Business users generally find that a dual floppy disk drive system is more efficient.

The central part of a small business accounting system is the chart of accounts which is the formal definition of the various asset, liability, capital, income and expense accounts. The Altair Business System General Ledger Package is based on a traditional dual entry accounting system. Although the General Ledger Package does allow a user considerable flexibility in establishing a chart of accounts, the system does require that the chart of account numbers for each of the account types (such as assets) be within a specified range. Assets are numbered 100 to 199, liabilities and capital 200 to 299, income 300 to 399 and expenses 400 to 999.

The General Ledger Package allows the user to control the printing format of the balance sheet and income statements by specifying accumulating levels and column printing specifications at the time the chart of accounts is established.

After the establishment of a chart of accounts, the General Ledger transactions are entered into the system from time to time during the month. At the end of each month, or whatever period is used by the particular business, the detailed general ledger is printed. This report provides a complete display of all of the monthly transactions grouped together by general ledger account numbers. The balance which had been previously retained in that account is displayed with the new current balance. The balance sheet and income statement can then be printed for that particular period. The income statement displays the total for the particular accounts

Continued

Software Notes from the ASDC

By John Hayes
General Manager, ASDC

The Altair User Group Software Library, located at ASDC headquarters in Atlanta, is a collection of programs and routines contributed by Altair computer users throughout the world. By the end of February, the Library contained more than 270 programs and routines, making it one of the largest user group libraries in existence.

The programs and routines in the User Group Library are usually written in machine or assembly language or in one of several versions of Altair BASIC.

Interested Altair computer users who submit programs and routines are given coupons which can be applied toward the purchase of other software in the User Group Library. Library programs are available, without any warranty, for a nominal copying/mailing charge.

Users should notify ASDC immediately about any program fixes which need to be made to programs in the library. We will send coupons for software credit to users who submit fixes. When notice of a fix is received, the information is put into the program file and distributed with each subsequent order for the program. Notices of fixes for more popular programs are published in COMPUTER NOTES.

We would like most of the programs in the Library to be available in machine readable form. So users are encouraged to send a paper tape or cassette to the ASDC. We will send a new cassette as well as software coupons to anyone who submits programs on cassette.

The Altair computer center in Denver and a number of other Altair computer users around the country are assisting the ASDC in putting many of the Library's programs and routines into machine readable forms. Groups of these programs will soon be published as a book, and machine readable paper tapes, cassettes and diskettes of the programs will also be available. The first of these sets will be available in late spring this year.

A brief description of each accepted program or routine is published each month in C.N. A description of each program is also published on a monthly update page for the Altair User Group Software Library Catalog. The update pages are distributed to regular subscribers of C.N. A current catalog as well as a one-year subscription to C.N. is given free to each new purchaser of an Altair computer. Other Altair computer users can obtain a current copy of the catalog for $5 from the ASDC.

Programs submitted to the User Group Library are automatically entered in the montly software contest. Prizes, in the form of credit towards the purchase of Altair equipment or software, are awarded for the best routines and programs.

Programs submitted to the Library are available to the public without restriction. However, the software author and the Altair User Group should be acknowledged for their work.

2-14-771--$2.00
Author: Baron L. Ader
Length: 27 bytes
Title: "Subroutine 'Text' for Altair 6800"

Machine or Assembly language routine allows output of text strings or Memory dumps to Teletype easily.

2-9-771--$2.00
Author: Peter Smart
Length: 16 lines
Title: "Game of Life" for Altair 6800

Game deals with the life in various cells on a rectangular grid.

2-16-771--$3.00
Author: Ron Santore
Length: 1.5 bytes, 8800 Assembly
Title: "WUMPUS"

An interesting game; the Wumpus you are hunting lives in a dodecahedron maze of 20 caves. Danger lurks from Wumpus, Superbats, and bottomless pits. Well documented with instructions for changing the I/O routines.

SOFTWARE

Altair General Ledger Package Insures Accurate Auditing

Continued

for that month as well as the accumulated year-to-date. The percentage allocation for both the current period and the year-to-date are also displayed.

The General Ledger Package is designed to insure auditing integrity. For example, printouts of all changes and additions to the chart of accounts are available to insure a visible audit trail. The system provides for the easy addition of a new chart of accounts. However, an existing account in the chart of accounts which has a balance cannot be deleted until that balance is taken to zero. Once a journal transaction has been entered in the detailed general ledger, it cannot be deleted. However, a reverse entry can be made, which has the effect of negating the dollar amount. But both the original entry and the reversed entry will show up on the general ledger detail report. Additional audit information is provided by the use of a source code and a reference description in each detailed journal transaction.

The General Ledger Package can be used as a stand-alone system independent of the other modules of the account package. It can also be used in conjunction with any of the other modules to form an integrated small business accounting system--the Payroll Package, the Accounts Receivable Package and the Accounts Payable Package--are designed to automatically provide input to the General Ledger at the end of each month. For example, when the user installs the Payroll Package, he defines a set of general ledger chart of account numbers that apply to the transactions which are passed from the Payroll Package to the General Ledger. Each month the General Ledger, under program control, "picks up" these transactions as detailed journal entries.

The General Ledger Package, like other components of the Altair Business System, comes with comprehensive documentation of the system and operator use of the package. Helpful guides are also provided to assist a user in establishing an automated bookkeeping and financial reporting system using the packages.

ARITHMETIC ERRORS "BUG" USERS

By Gale W. Schonfeld

Many Altair BASIC users have recently noticed that the answers they get from BASIC arithmetic computations differ significantly from the answers they expect. Although some users attribute these discrepancies to bugs in BASIC, they are merely errors which are inherent in almost all arithmetic operations performed by calculators or computers. These errors occur in binary, decimal or binary coded decimal arithmetic as well as in fixed and floating point. They are unavoidable but can be comprehended and, to some extent, predicted.

There are two main categories of arithmetic error: (1) human and (2) round-off. I'll by-pass human error, since we all know humans are pretty close to perfect, and go straight to round-off errors. Round-off is used constantly in mathematics, for long numbers, repeating decimal fractions, non-terminating decimal fractions, etc.

The procedure for rounding-off is: (1) determine the number of significant digits desired, (2) discard the non-significant digits, then (3) if the discarded digits are less than half a unit (usually determined by the left-most digit of the discard, i.e. 1/2 = 5 in the decimal system), leave the right-most significant digit unchanged. If the discarded digits are more than half a unit, then round the right-most significant digit up by adding 1. If the discarded digits equal half a unit, then round the right-most significant digit to the next greater even digit (i.e. add one or leave the figure unchanged so that it is even). (Some mathematicians prefer to round-off to the nearest odd digit. The major concern is to be consistent.)

The following examples show how round-off can eventually cause errors in the final answer. This is particularly true in computers where the number of digits is limited. If the number being used is longer than allowed, round-off is required either by the computer or the user.

EXAMPLE:

Number	Rounded to 4 Significant Digits
23.47821	23.48
00.02671154	.02671
00.12345	.1234
00.12355	.1236

Round these repeating decimal fractions:

1/3 = 0.33333333...	.3333
2/3 = 0.66666666...	.6667

Continued

END PROGRAMMING HUMDRUM WITH YAHTSEE™

By: Jim Gerow

Gerow works in New York as a software designer and programmer on mission similators for the Nassau space shuttle.

Break your computer's usual programming routine by challenging it to a game of Yahtzee™.

Based on the popular E.S. Lowe dice game, this program has extra logic built in so that the computer can move as one of the players. In fact, you can set it up to play itself. One note of caution: before you teach your Altair computer how to play Yahtzee, practice! My computer beat me by well over 100 points the first time I played.

The object of the game is to accumulate the highest score. There are thirteen scoring categories ranging from aces to sixes and from three-of-a-kind to chance. Each player can score in any of the unused categories with up to three rolls of the five dice. The computer prints the total after each player's turn and the complete scoresheet after the game.

If you're a novice Yahtzee player, you can ask the computer to list your possibilities for scoring. When you graduate to intermediate status, see how many times you have to ask the computer for help. For the experts, get out your Yahtzee score pads and don't ask for any help.

If you have the gang over, set sense switch 15 and the computer will print out the scores after each round. But if there's no one else around, you can even let the Altair computer play by itself. Say "YES" to letting it play and answer "0" to the number of human players. Notice that if you set the "human" player's name(s) to "Altair," the computer will think it is its turn and will play.

Remember, don't ignore your computer while you rid the galaxy of the deadly Klingon menace or blast a new lunar crater 200 feet deep. Let it join the fun.

(NOTE: This program is terminal-dependent. If your terminal is different from mine, change lines 9510, 9515 and 9530 to suit your system.)

```
10 REM
20 REM  YAHTZEE  GAME    --PROGRAMMED BY JIM GEROW 10/76
30 REM         >> WRITTEN IN MITS 8K BASIC <<
40 REM  *** INITIALIZATION SECTION ***
50 CLEAR 250               : REM  ALLOCATE STRING SPACE
60 DEF FNA(X)=INT(5.99999*RND(1))+1  :REM  DEFINE DICE ROLL FUNCTION
70 REM *** PRINT TITLE ***
80 PRINT : PRINT : PRINT TAB(20); "YAHTZEE   GAME " : PRINT : PRINT
90 REM *** OUTPUT SENSE SWITCH OPTIONS ***
100 PRINT TAB(16); "SENSE SWITCH OPTIONS:" : PRINT
110 PRINT TAB(15); "SSW #"; TAB(25); "MEANING WHEN SET"
120 PRINT
130 PRINT TAB(16); "15"; TAB(25);
140 PRINT "PRINT TOTALS AFTER THIS ROUND"
150 PRINT : PRINT
250 REM *** CHECK IF ALTAIR PLAYS ***
260 INPUT "DO YOU WANT ME TO PLAY"; T$
270 T$=LEFT$(T$,1)      : REM EXTRACT FIRST LETTER OF REPLY
280 IF NOT (T$="Y" OR T$="N") THEN 260  : REM CHECK IF VALID Y OR N
290 N1=1 : IF T$="Y" THEN N1=0 : REM SET FLAG TO 0 IF ALTAIR PLAYS
300 REM *** INPUT NUMBER OF PLAYERS ***
310 INPUT "NUMBER OF HUMAN PLAYERS"; N2
320 IF N2=0 AND N1=1 THEN END : REM END IF NO PLAYERS
330 IF N2<>0 THEN 360      : REM  GO DIMENSION FOR PLAYERS
340 N2=1 : N1=1       : REM  SET UP FOR ALTAIR ONLY
350 REM *** DIMENSIONING ***
360 DIM NAME$(N2)       : REM  PLAYERS' NAMES ARRAY
370 DIM SC(19,N2)       : REM  PLAYERS' SCORES ARRAY
380 DIM T(N2)           : REM  TURN ORDER & WORK ARRAY
390 DIM D(5),D1(5),D2(5) : REM  DICE ARRAYS
400 DIM LB$(19),S(13) : REM  SCORING TITLES AND VALUES ARRAYS
410 REM *** ERASE SCORE SHEET ***
420 FOR I=1 TO 19 : FOR J=0 TO N2 : SC(I,J)=-1 : NEXT : NEXT
430 REM *** SET UP PLAYERS' NAMES ***
440 IF N1=0 THEN NAME$(0)="ALTAIR"
450 REM IF ONLY ALTAIR PLAYS, GO START
460 IF T$="Y" AND N1=1 THEN NAME$(1)="ALTAIR" : GOTO 1000
470 FOR I=1 TO N2       : REM  INPUT PLAYERS' NAMES
480 PRINT "NAME OF PLAYER  #"; I;
490 INPUT ""; NAME$(I) : NEXT : PRINT
495 IF N1=N2 THEN PRINT " >>> PRESS SPACE TO ROLL DICE <<<": GOTO 1000
500 REM *** ROLL OF DICE FOR TURN ORDER ***
510 PRINT : PRINT "   ROLL OF DICE FOR TURN ORDER" : PRINT
520 PRINT " >>> PRESS SPACE BAR TO ROLL DICE <<< " : PRINT
530 IF N1<>0 THEN 570   : REM  IF ALTAIR DOESN'T PLAY, SKIP
540 PRINT "ALTAIR'S ROLL IS:"
550 REM * ROLL DICE, PRINT AND STORE TOTALS
560 GOSUB 9000 : PRINT "   TOTAL "; TTL : T(0)=TTL
570 PRINT : FOR I=1 TO N2     : REM  OTHER PLAYERS' ROLLS
580 GOSUB 9500    : REM  PRINT THE PLAYER'S NAME
```

```
590 REM * ROLL DICE, PRINT AND STORE TOTALS
600 GOSUB 9000 : PRINT "    TOTAL "; TTL : T(I)=TTL
610 PRINT : NEXT
620 REM *** SORT TOTALS FOR TURN ORDER ***
630 J=0 : FOR I=N1 TO N2-1
640 IF T(I)>=T(I+1) THEN 680
650 T=T(I+1) : T(I+1)=T(I) : T(I)=T   : REM SWAP TOTAL & NAME
660 T$=NAME$(I+1) : NAME$(I+1)=NAME$(I) : NAME$(I)=T$
670 J=1        : REM  SET FLAG TO INDICATE A SWAP
680 NEXT : IF J<>0 THEN 630  : REM  IF A SWAP WAS MADE, REPEAT
690 PRINT : PRINT "  ORDER OF TURNS:" : PRINT
700 FOR I=N1 TO N2
710 PRINT (I-N1+1); "  "; NAME$(I)
720 NEXT
990 REM *** PRINT SCORING TITLES ***
1000 PRINT : PRINT : PRINT "  *** SCORING TABLE ***" : PRINT
1010 PRINT "ROW       DESCRIPTION" : PRINT
1020 PRINT "  UPPER HALF SCORING" : PRINT
1030 LB$( 1)=" ACES" : PRINT " 1  --  "; LB$(1)
1040 LB$( 2)=" TWOS" : PRINT " 2  --  "; LB$(2)
1050 LB$( 3)=" THREES" : PRINT " 3  --  "; LB$(3)
1060 LB$( 4)=" FOURS" : PRINT " 4  --  "; LB$(4)
1070 LB$( 5)=" FIVES" : PRINT " 5  --  "; LB$(5)
1080 LB$( 6)=" SIXES" : PRINT " 6  --  "; LB$(6)
1090 PRINT : PRINT "  LOWER HALF SCORING" : PRINT
1100 LB$( 7)=" 3 OF A KIND" : PRINT " 7  --  ";LB$(7)
1110 LB$( 8)=" 4 OF A KIND" : PRINT " 8  --  ";LB$(8)
1120 LB$( 9)=" FULL HOUSE" : PRINT " 9  --  ";LB$(9)
1130 LB$(10)=" SM. STRAIGHT" : PRINT "10  --  ";LB$(10)
1140 LB$(11)=" LG. STRAIGHT" : PRINT "11  --  ";LB$(11)
1150 LB$(12)=" YAHTZEE" : PRINT "12  --  ";LB$(12)
1160 LB$(13)=" CHANCE" : PRINT "13  --  ";LB$(13)
1170 PRINT : PRINT
1180 PRINT " DICE ARE LISTED:" : PRINT "A B C D E"
1190 PRINT
1200 REM  *** START OF LOOP FOR THE PLAYING ROUND ***
1210 FOR R=1 TO 13 : PRINT : PRINT "ROUND "; R : PRINT
1220 FOR PL=N1 TO N2  :REM PLAYER NUMBER
1225 J1=0  :REM INITIALIZE TO FIRST ROLL
1230 IF NAME$(PL)<>"ALTAIR" THEN 1270  :REM CHECK IF ALTAIR'S TURN
1240 REM *** ALTAIR'S TURN ***
1250 PRINT "ALTAIR'S TURN": GOSUB 9000: L=0: GOSUB 5000: GOTO 1490
1260 REM *** PLAYER'S TURN ***
1270 I=PL : GOSUB 9500  :REM PRINT PLAYER'S NAME & WAIT FOR SPACE
1280 GOSUB 9000 : GOSUB 8000 :REM ROLL AND TALLY POSSIBILITIES
1290 J1=1      :REM SET FOR FIRST ROLL
1300 REM *** SUCCESSIVE ROLLS ***
1310 PRINT "ENTER: P FOR POSSIBILITIES, R FOR ANOTHER ROLL, S TO SCORE"
1311 INPUT T$ : T$=LEFT$(T$,1)
1312 IF T$="P" THEN GOSUB 8640 : GOTO 1310
1320 IF T$="S" THEN 1460 :REM *** GO SCORE THIS ROLL
1330 IF T$<>"R" THEN 1310 :REM REPEAT QUESTION
1340 INPUT "DICE TO RE-ROLL"; T$
1350 J=0 : IF T$="" THEN 1340
1360 J=J+1 :REM LOOP TO RE-ROLL DICE
1370 T1$=MID$(T$,J,1) : IF T1$="" THEN 1420 :REM CHECK IF DONE
1380 T=ASC(T1$)-64 : IF T<1 OR T>5 THEN 1360 :REM VALIDATE
1390 PRINT CHR$(T+64); " ";  :REM ECHO EACH VALID DIE
1400 D(T)=FNA(1)   :REM USE FNA TO RE-ROLL DIE
1410 GOTO 1360       :REM CHECK IF ANY MORE TO RE-ROLL
1420 TTL=D(1)+D(2)+D(3)+D(4)+D(5)   :REM CALCULATE TOTAL
1430 PRINT : GOSUB 9010 : GOSUB 8000 :REM PRINT DICE & CHECK CHOICES
1440 J1=J1+1 : IF J1<3 THEN 1310 :REM CHECK IF ANOTHER ROLL WANTED
1444 PRINT "ENTER: P FOR POSSIBILITIES, S TO SCORE"
1445 INPUT T$ : T$=LEFT$(T$,1)
1446 IF NOT(T$="P" OR T$="S") THEN 1445
1447 IF T$="P" THEN GOSUB 8640 :REM PRINT POSSIBILITIES
1450 REM *** SCORING ***
1460 PRINT : INPUT "WHAT ROW # DO YOU WISH TO SCORE "; T
1470 IF T<1 OR T>13 THEN 1460  :REM VALIDATE
1480 IF SC(T,PL)<>-1 THEN 1460 :REM CHECK IF FREE
1490 SC(T,PL)=S(T)     :REM ENTER SCORE IN TABLE
1500 REM *** OUTPUT POINTS SCORED ***
1510 PRINT S(T); " POINTS ARE SCORED FOR"; LB$(T)
1520 REM *** TOTAL UPPER HALF ***
1530 T=0 : FOR I=1 TO 6 : IF SC(I,PL)>0 THEN T=T+SC(I,PL)
1540 NEXT : SC(14,PL)=T
1550 REM *** TEST FOR BONUS ***
1560 SC(15,PL)=0 : IF T>62 THEN T=T+35 : SC(15,PL)=35
1570 SC(16,PL)=SC(14,PL)+SC(15,PL)  :REM CALCULATE UPPER HALF TOTAL
1580 REM *** TOTAL LOWER HALF ***
1590 T=0 : FOR I=7 TO 13 : IF SC(I,PL)>0 THEN T=T+SC(I,PL)
1600 NEXT : SC(17,PL)=T : SC(18,PL)=SC(16,PL)
1610 REM *** CALCULATE GRAND TOTAL ***
1620 T=T+SC(18,PL) : SC(19,PL)=T
1630 PRINT NAME$(PL); "'S TOTAL IS NOW "; T
1640 PRINT
1650 NEXT    :REM END OF PLAYER'S TURN, DO ANOTHER
1660 REM *** END OF ROUND ***
1670 REM *** CHECK IF PARTIAL SCORING WANTED ***
1680 IF (INP(255) AND 128) = 128 THEN 1710
1690 IF R<>13 THEN 2000   :REM IF END OF GAME, PRINT TOTALS
1700 REM *** PRINT OUT SCORESHEET ***
1710 IF R=13 THEN PRINT : PRINT TAB(20); "*** FINAL SCORING ***"
1712 PRINT : PRINT : FOR J=N1 TO N2 : I=J-N1+1
1720 PRINT "PLAYER  #"; I; "   "; NAME$(J)
1730 NEXT : PRINT : PRINT
1740 PRINT "  TOTALS  " : PRINT : PRINT
1750 FOR I=N1 TO N2 : J=I-N1+1
1760 T1$=STR$(J) : T$=RIGHT$(T1$,(LEN(T1$)-1))
```

Continued

ARITHMETIC ERRORS "BUG" USERS

Continued

The fact that computers must round numbers to a certain number of digits is only half the problem.

The relation between software and hardware is where the other part of the arithmetic problem arises. In order to do arithmetic operations on the computer (other than front panel toggling in binary), software must be written to convert decimal numbers to binary. The arithmetic is then carried out in binary and converted back to decimal. This decimal to binary conversion often causes problems since not all decimal numbers can be represented exactly in binary or vice-versa.

In Table 1 the conversion is calculated by doubling the decimal fraction, obtaining a 0 or 1 to the left of the decimal point in the product and adding 0 or 1 to the right of the binary number. Conversion continues until the product is equal to 1.0 or the desired number of significant digits is calculated.

I used Thomas Bartee's DIGITAL COMPUTER FUNDAMENTALS for the methods of conversion on differing number bases in Table 1. Although the book is mainly hardware oriented, it contains some very valuable information relating to software fundamentals.

Table 1

DECIMAL	to	BINARY
0.1_{10}		
2 x 0.1 = 0.2	to	.0
2 x 0.2 = 0.4		.00
2 x 0.4 = 0.8		.000
2 x 0.8 = 1.6		.0001
2 x 0.6 = 1.2		.00011
2 x 0.2 = 0.4		.000110
2 x 0.4 = 0.8		.0001100
2 x 0.8 = 1.6		.00011001
2 x 0.6 = 1.2		$.0001\overline{1001}1\ldots_2$
0.2_{10}		
2 x 0.2 = 0.4	to	.0
2 x 0.4 = 0.8		.00
2 x 0.8 = 1.6		.001
2 x 0.6 = 1.2		.0011
2 x 0.2 = 0.4		.00110
2 x 0.4 = 0.8		.001100
2 x 0.8 = 1.6		.0011001
2 x 0.6 = 1.2		$.0011\overline{0011}\ldots_2$

(Note: The bar over the last four binary digits means that those digits repeat indefinitely.)

DECIMAL to BINARY

$.0011\overline{0011}\ldots_2$

$$2^{-4} + 2^{-5} + 2^{-8} + 2^{-9} = 0.099609375_{10}$$

$.0011\overline{0011}\ldots_2$

$$2^{-3} + 2^{-4} + 2^{-7} + 2^{-8} = 0.19921875_{10}$$

Continued

ARITHMETIC ERRORS "BUG" USERS

Continued

The following math problem shows two different decimal methods, how to convert to binary, perform the arithmetic and then convert back to decimal:

PROBLEM: 2/3 * 9 = X

I. Solve by conventional cross-multiplication using decimal arithmetic:

$\frac{2}{1} * 3 = 6$ OR $\frac{2}{3} * 9 = \frac{18}{3} = 6$

Therefore X = 6

II. Solve according to Table of Precedence (see Altair BASIC Reference Manual) using decimal arithmetic and round-off:

2/3 * 9 = .6667 * 9 = 6.0003

Therefore X - 6.0003

(Note: If you carried the repeating decimal of 2/3 = 0.66666..., out to more significant digits before rounding-off, the error would be less, but it would still be there.)

III. Solve according to Table of Precedence using binary arithmetic:

A. First convert the problem to binary:

$2_{10} = 10_2$

$3_{10} = 11_2$

$9_{10} = 1001_2$

Restated, the problem is:

$10_2/11_2 * 1001_2 = X$

B. Divide:

$10_2/11_2 = .1010_2$

$.1\overline{010}$ ... 2

$11_2 \quad 10.0000_2$

C. Multiply:

$.1010_2 * 1001_2 =$

101.1010_2

$$\begin{array}{r} 1001_2 \\ \times \quad .1010_2 \\ \hline 10010 \\ 0000 \\ 1001 \\ \hline 101.1010_2 \end{array}$$

Therefore X = 101.1010_2

Continued

END PROGRAMMING HUMDRUM WITH YAHTSEE™

Continued

```
1770 PRINT TAB(J*5+19); "#"; T$; : NEXT : PRINT
1780 REM *** PRINT UPPER HALF SCORING ***
1790 FOR I=1 TO 6
1800 GOSUB 9600 : NEXT
1810 LB$(14)=" TOTAL OF ABOVE"
1820 LB$(15)=" BONUS IF >= 63"
1830 LB$(16)=" TOTAL OF UPPER HALF"
1840 LB$(17)=" TOTAL OF LOWER HALF"
1850 LB$(18)=LB$(16)
1860 LB$(19)=" GRAND TOTALS"
1870 REM *** PRINT SUBTOTAL AND BONUS ***
1880 PRINT : FOR I=14 TO 16 : GOSUB 9600 : NEXT
1890 REM *** PRINT LOWER HALF SCORING ***
1900 PRINT : PRINT : FOR I=7 TO 13 : GOSUB 9600 : NEXT
1910 REM *** PRINT TOTALS ***
1920 PRINT : PRINT : FOR I=17 TO 19 : GOSUB 9600 : NEXT
1990 REM *** END OF ROUND ***
2000 NEXT
2010 REM *** END OF GAME ***
2020 END
4990 REM ***ALTAIR'S ROLL AND SCORING ***
5000 GOSUB 8000 : L=L+1   :REM CALCULATE THE POSSIBILITIES
5010 IF S(12)<>0 THEN T=12 : RETURN  :REM IF YAHTZEE, GO SCORE IT
5020 IF S(11)<>0 THEN T=11 : RETURN  :REM IF LG. STRAIGHT, SCORE IT
5030 IF S(9)<>0 THEN T=9 : RETURN  :REM IF FULL HOUSE, GO SCORE IT
5035 IF SF=0 THEN 5210  :REM IF NO STRAIGHT, CHECK ELSEWHERE
5040 IF L<>3 AND SC(11,PL)=-1 THEN 5080  :REM TRY FOR LG. STRAIGHT
5050 IF S(10)=0 THEN 5210  :REM IF SM. STRAIGHT USED, CHECK OTHER ROWS
5060 IF SC(11,PL)>=0 THEN T=10 : RETURN :REM SCORE SM STR. IF LG. USED
5070 IF L=3 THEN T=10 : RETURN :REM IF LAST ROLL, SCORE THE SM STRAIGHT
5080 I=0   :REM  DETERMINE DIE TO ROLL FOR LG. STRAIGHT
5090 I=I+1 : IF D1(I)=D1(I+1) THEN K=D1(I) : GOTO 5130
5100 IF I<4 THEN 5090
5110 IF D1(1)=1 AND D1(2)<>2 THEN K=1 : GOTO 5130
5120 K=6
5130 PRINT "ALTAIR'S NEXT ROLL"
5140 I=0
5150 I=I+1 : IF D(I)=K THEN 5180
5160 IF I<5 THEN 5150
5170 STOP  :REM ERROR IF NOTHING MATCHES
5180 D(I)=FNA(1)  :REM RE-ROLL THE DIE
5190 TTL=D(1)+D(2)+D(3)+D(4)+D(5)  :REM CALCULATE TOTAL
5200 GOSUB 9010 : GOTO 5000 :REM OUTPUT DICE, TRY THIS ROLL
5210 IF S(8)=0 THEN 5280  :REM CHECK IF 4 OF A KIND
5220 IF L<>3 THEN 5260 :REM IF NOT LAST ROLL, CHECK DICE TO RE-ROLL
5230 IF TTL>18 THEN 5250 :REM IF HIGH TOTAL, SCORE IN 4 OF A KIND
5240 IF SC(D1(2),PL)=-1 THEN T=D1(2) : RETURN :REM ELSE TRY UPPER HALF
5250 T=8 : RETURN :REM IF ALL ELSE FAILS, SCORE IN 4 OF A KIND
5260 IF D1(1)=D1(2) THEN K=D1(5) : GOTO 5130 :REM TRY FOR YAHTZEE
5270 K=D1(1) : GOTO 5130  :REM TRY FOR YAHTZEE
5280 IF S(7)=0 THEN 5370  :REM IF NOT 3 OF A KIND, TRY UPPER HALF
5290 J=3 : IF L<>3 THEN 5320  :REM TRY FOR 3 OF A KIND
5300 IF SC(D1(3),PL)=-1 THEN T=D1(3) : RETURN :REM SCORE IN UPPER HALF
5310 T=7 : RETURN  :REM SCORE IN 3 OF A KIND
5320 PRINT "ALTAIR'S NEXT ROLL"
5330 FOR I=1 TO 5  :REM TRY TO ADD TO 3 OF A KIND
5340 IF D(I)=D1(J) THEN 5360  :REM IF MATCHES, DON'T ROLL IT
5350 D(I)=FNA(1)  :REM RE-ROLL THE DIE
5360 NEXT : GOTO 5190 :REM CALCULATE TOTAL AND START PRINT RESULTS
5370 I=7 : IF L<>3 THEN 5410  :REM IF NOT LAST ROLL, TRY AGAIN
5380 I=I-1 : IF S(I)/I>=3 THEN T=I : RETURN :REM TRY TO FIT IN UPPER
5390 IF I>1 THEN 5380
5400 GOTO 5520  :REM NO LUCK, FIND SOMEPLACE TO SCORE
5410 J=5 : K=0   :REM SET UP FOR NEXT ROLL
5420 J=J-1 : IF D1(J)<>D1(J+1) THEN 5450  :REM TEST FOR A PAIR
5430 T=D1(J)
5440 K=1 : IF SC(D1(J),PL)=-1 THEN 5320  :REM PAIR, CHECK IF ROW FREE
5450 IF J>1 THEN 5420  :REM LOOP TO TRY TO FIND A PAIR
5460 IF SF=1 AND SC(11,PL)=-1 THEN 5080 :REM CHECK STRAIGHT IF NO PAIR
5470 J=0
5480 J=J+1 : IF SC(D1(J),PL)=-1 THEN 5320 :REM TRY FOR FREE ROW TO FILL
5490 IF J<5 THEN 5480  :REM KEEP LOOPING
5500 PRINT "ALTAIR'S NEXT ROLL"
5510 GOSUB 9000: GOTO 5000 :REM WHEN ALL ELSE FAILS RE-ROLL ALL 5
5515 REM *** IF LAST ROLL, FIND SOME PLACE TO SCORE ***
5520 IF SC(13,PL)=-1 THEN T=13 : RETURN :REM SCORE TOTAL IN CHANCE
5530 IF SC(12,PL)=-1 THEN T=12 : RETURN :REM SCORE A 0 IN YAHTZEE
5540 IF SC(11,PL)=-1 THEN T=11 : RETURN  :REM SCORE A 0 IN LG. STRAIGHT
5550 IF SC(9,PL)=-1 THEN T=9 : RETURN  :REM SCORE A 0 IN FULL HOUSE
5560 IF SC(10,PL)=-1 THEN T=10 : RETURN :REM SCORE A 0 IN SM. STRAIGHT
5570 IF SC(8,PL)=-1 THEN T=8 : RETURN  :REM SCORE A 0 IN 4 OF A KIND
5580 T=0  :REM TRY TO SCORE IN UPPER HALF
5590 T=T+1
5600 IF SC(T,PL)=-1 THEN RETURN  :REM A FREE ROW, SCORE HERE
5610 IF T<=6 THEN 5590  :REM LOOP TO FIND A FREE ROW
5620 STOP  :REM ERROR, NO FREE ROWS
5630 END
7990 REM *** TALLY SCORING AND PRINT CHOICES ***
8000 PRINT : FOR N=1 TO 5 : D1(N)=D(N) : NEXT :REM INITIALIZE WORK ARRA
8005 SF=0
8010 J=0 : FOR N=1 TO 4  :REM PUT DICE IN NUMERICAL ORDER
8020 IF D1(N)<=D1(N+1) THEN 8040
8030 J=1 : T=D1(N) : D1(N)=D1(N+1) : D1(N+1)=T
8040 NEXT : IF J<>0 THEN 8010
8050 FOR J=1 TO 13 : S(J)=0 : NEXT  :REM CLEAR SCORES FOR THIS TURN
8060 I=2  :REM  CHECK FOR 5 OF A KIND
```

Continued

ARITHMETIC ERRORS "BUG" USERS

Continued

D. Convert back to decimal:

$.1010_2 = .625_{10}$

$1001_2 = 9_{10}$

$101.1010_2 = 5.625_{10}$

$.625 * 9 = 5.625$

$X = 5.625$

Now let's evaluate the above problem.

Method I X = 6
Method II X = 6.0003
Method III X = 5.625

If we round these results to two significant digits:

$X_{(I)} = 6.0$

$X_{(II)} = 6.0$

$X_{(III)} = 5.6$

These answers show that round-off during calculations and on final answers can cause unforeseen errors. For more information about round-off and other topics, see the bibliography at the end of the article.

Just remember that each computer system handles round-off according to its own rules. Altair BASIC follows the rules on pp. 10-19 of the Altair BASIC 4.0 Reference Manual.

For the sake of time, the example in Table 1 uses 'short' numbers. So the round-off errors are large compared to what they would be if carried out to more significant digits within the computer using single or double precision BASIC.

Once the programmer knows how the computer handles computations and where numberical errors can arise, it's easier to accept these potential errors.

Bibliography:

(1) Actor, Forman S.: NUMERICAL METHODS THAT WORK, Harper & Row, p. 311.

(2) Bartee, Thomas C. (1966): DIGITAL COMPUTER FUNDAMENTALS, Second Edition, McGraw-Hill, Inc., New York, Chapter 3.

(3) Henrici, Peter (1964): ELEMENTS OF NUMERICAL ANALYSIS, John Wiley & Sons, Inc., New York, Chapters 15 and 16.

(4) Hildebrand, F. B. (1956): INTRODUCTION TO NUMERICAL ANALYSIS, McGraw-Hill, Inc., New York, Chapter 1.

(5) Noble, Ben (1964): NUMERICAL METHODS I, ITERATION, PROGRAMMING AND ALGEBRAIC EQUATIONS, Oliver and Boyd, Edinburgh and London, Chapter 1.

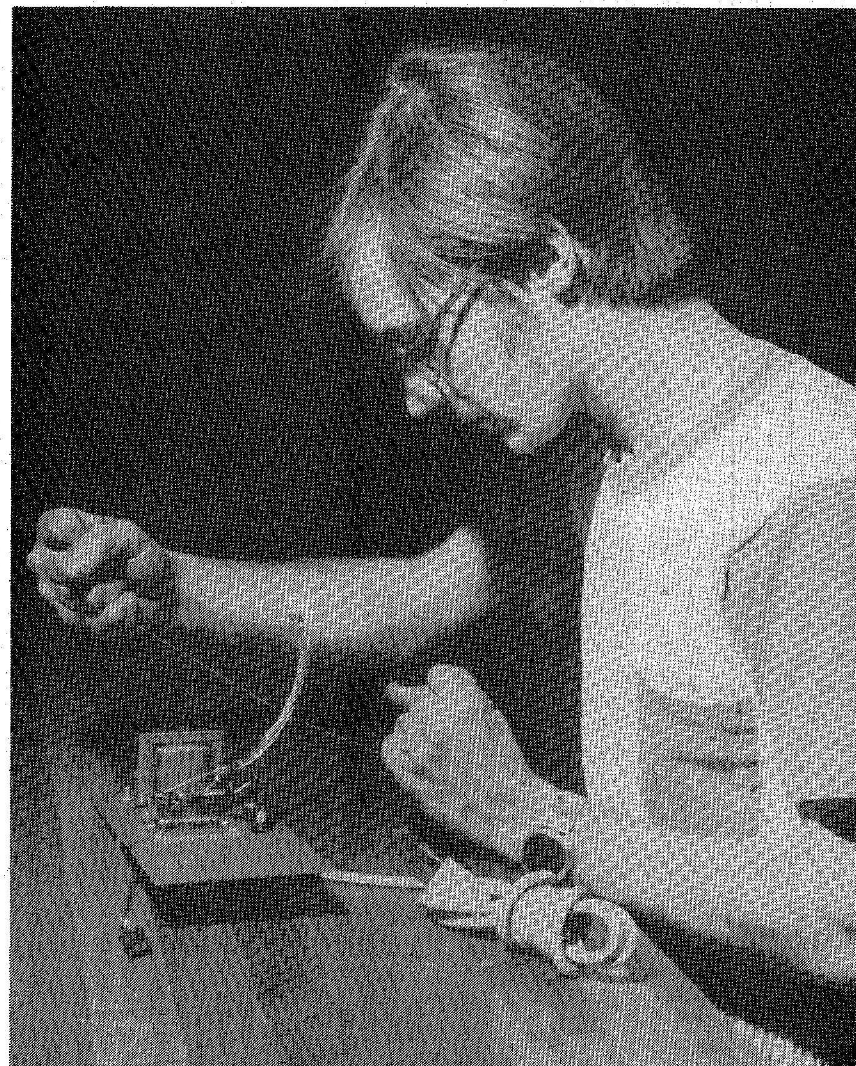
(6) Ralston, Anthony (1965): A FIRST COURSE IN NUMERICAL ANALYSIS, McGraw-Hill, Inc., New York, Chapter 1.

END PROGRAMMING HUMDRUM WITH YAHTSEE™

Continued

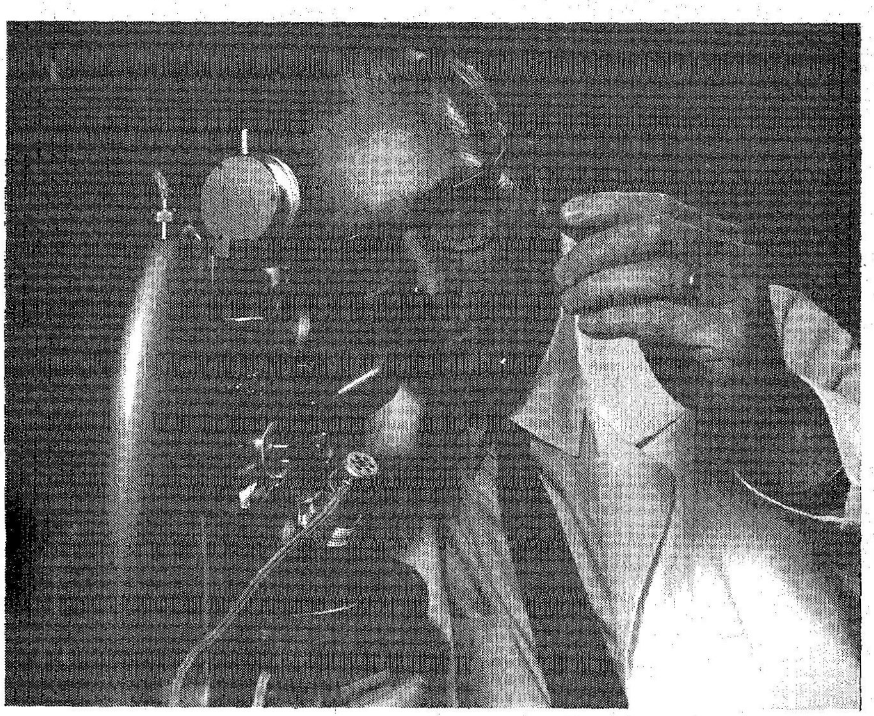
```
8070 IF D1(1)<>D1(I) THEN 8170
8080 I=I+1: IF I<6 THEN 8070
8085 REM *** FIVE OF A KIND ***
8090 IF SC(12,PL)=-1 THEN S(12)=50 :REM SCORE YAHTZEE IF FREE
8100 IF SC(8,PL)=-1 THEN S(8)=TTL  :REM SCORE 4 OF A KIND IF FREE
8110 IF SC(7,PL)=-1 THEN S(7)=TTL  :REM SCORE 3 OF A KIND IF FREE
8120 IF SC(13,PL)=-1 THEN S(13)=TTL  :REM SCORE CHANCE IF FREE
8125 REM ***SCORE IN UPPER HALF ***
8130 FOR I=1 TO 5
8140 IF SC(D1(I),PL)<>-1 THEN 8160  :REM CHECK IF ROW IS FREE
8150 S(D1(I))=S(D1(I))+D1(I)      :REM SUM VALUE OF DIE
8160 NEXT : RETURN
8170 I=2    :REM TEST FOR FOUR OF A KIND
8180 IF D1(1)<>D1(I) THEN 8210
8190 I=I+1 : IF I<5 THEN 8180
8200 GOTO 8240   :REM 4 OF A KIND, GO SCORE IT
8210 I=3   :REM KEEP TESTING FOR 4 OF A KIND
8220 IF D1(2)<>D1(I) THEN 8250
8230 I=I+1: IF I<6 THEN 8220
8240 K=4 : GOTO 8100  :REM  SCORE 4 OF A KIND
8250 IF D1(1)=D1(2) AND D1(1)=D1(3) THEN 8310 ;REM CHK FOR 3 OF A KIND
8260 IF D1(2)=D1(3) AND D1(2)=D1(4) THEN 8310 :REM CHK FOR 3 OF A KIND
8270 IF NOT(D1(3)=D1(4) AND D1(3)=D1(5)) THEN 8330 :REM SKIP IF NOT
8280 IF D1(1)<>D1(2) THEN 8320  :REM CHECK FOR A FULL HOUSE
8290 IF SC(9,PL)=-1 THEN S(9)=25 : GOTO 8320  :REM SCORE FULL HOUSE
8300 GOTO 8320
8310 IF D1(4)=D1(5) THEN 8290  :REM CHECK FOR FULL HOUSE
8320 K=3 : GOTO 8110   :REM SCORE OTHER POSSIBILITIES
8325 REM *** CHECK FOR LARGE STRAIGHT ***
8330 FOR I=1 TO 4 : IF D1(5)=D1(I)+5-I THEN NEXT : GOTO 8350
8340 GOTO 8360     :REM NOT A LG. STRAIGHT
8350 IF SC(11,PL)=-1 THEN S(11)=40 : GOTO 8440 :REM SCORE LG.&SM.STR.
8355 REM *** SET UP FOR SM. STRAIGHT TEST ***
8360 FOR I=1 TO 5 : D2(I)=0 : NEXT : D2(1)=D1(1) : J=1
8370 FOR I=2 TO 5
8380 IF D1(I)=D2(J) THEN 8400
8390 J=J+1 : D2(J)=D1(I)
8400 NEXT : IF J<4 THEN 8120   :REM IF NOTHING FITS TRY CHANCE
8405 REM *** CHECK FOR A SMALL STRAIGHT ***
8410 FOR I=1 TO 3 : IF D2(4)=D2(I)+4-I THEN NEXT : GOTO 8440
8420 FOR I=2 TO 4 : IF D2(5)=D2(I)+5-I THEN NEXT : GOTO 8440
8430 GOTO 8120
8440 SF=1 : IF SC(10,PL)=-1 THEN S(10)=30 : GOTO 8120 :REM SCOR SM STR
8450 GOTO 8120
8630 REM *** PRINT SCORING CHOICES ***
8640 J=0 : FOR I=1 TO 13
8650 IF SC(I,PL)<>-1 THEN 8710  :REM IF ALREADY FILLED, SKIP
8660 IF J1>2 THEN 8680  :REM IF LAST ROLL, PRINT ALL CHOICES
8670 IF S(I)=0 THEN 8710 :REM IF NOT LAST ROLL, DON'T PRINT ZEROS
8680 PRINT TAB(J); "#"; RIGHT$(STR$(I),(LEN(STR$(I))-1));
8690 PRINT LB$(I); "("; S(I); ")";
8700 J=J+25 : IF J>51 THEN J=0 : PRINT    :REM ADJUST TAB
8710 NEXT : IF J=0 THEN RETURN
8720 PRINT : RETURN
8990 REM *** DICE ROLL, PRINT, AND TOTAL SUBROUTINE ***
9000 TTL=0 : FOR N=1 TO 5 : D(N)=FNA(1): TTL=TTL+D(N) : NEXT
9010 FOR N=1 TO 5 : PRINT D(N); : NEXT : RETURN
9490 REM *** PLAYER'S TURN SUBROUTINE ***
9500 PRINT "READY FOR "; NAME$(I); "'S TURN"
9510 N=-1 : J=INP(1) :REM RANDOMIZE WHILE WAITING FOR SPACE
9515 IF (INP(1) AND 127) = 32 THEN WAIT 0,1,1 : RETURN
9520 N=N-2 : IF N<-4095 THEN N=-1  :REM  GENERATE ODD NEGATIVE INTEGER
9530 J=INP(1) AND 127 : IF J<>32 THEN 9520 :REM WAIT FOR SPACE
9540 N=RND(N) :  RETURN   :REM SET NEW RANDOM "SEED" AND RETURN
9590 REM *** PRINT SCORING LINE SUBROUTINE ***
9600 PRINT LB$(I);  :REM  PRINT LABEL
9610 FOR J=N1 TO N2  :REM FOR EACH PLAYER PRINT THE SCORE
9620 IF SC(I,J)=-1 THEN PRINT TAB((J-N1+1)*5+19); "  ";: GOTO 9640
9630 PRINT TAB((J-N1+1)*5+19); SC(I,J);
9640 NEXT : PRINT : RETURN
OK
```

STUDENTS FIND ALTAIR 680B KIT

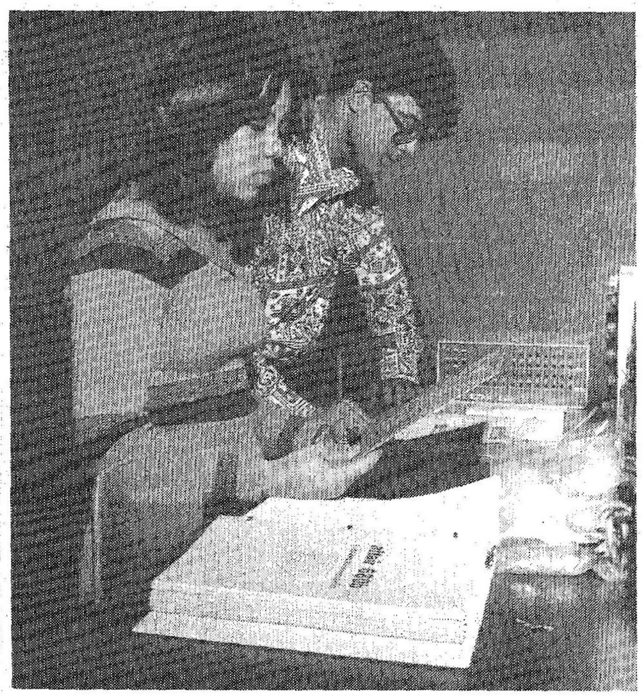


Power supply wires are laced to make trouble-free cable connections to the main circuit board.

EASY TO ASSEMBLE



James Gupton, former Vice-President of International CRT Corp., now teaches electronics at the Union County Career Center, Monroe, North Carolina.



Students check Altair 680b PC Board and assembly instructions.

By James Gupton

Gupton is a free-lance writer and an electronics teacher at the Union County Career Center in North Carolina. This is the first in a series of articles on his students' experiences with an Altair 680b computer.

In spite of a growing emphasis in computer publications upon the simplicity of microcomputers, many people still mistakenly believe that only an electronics expert can construct and operate a microcomputer. But with the aid of kit instructions and diagrams, my class of high school electronics students, who had no previous microcomputer training, built an Altair 680b computer in less than a week.

The class I teach at the Union County Career Center is part of a two-year electronics course designed to acquaint students with the fundamentals of electronics and to give them an opportunity to do independent research and development projects. I'm currently in the process of expanding the course to include instruction in computer hardware and software. The electronics course is just one of 10 different occupational trade courses now offerred to juniors and seniors of Union County high schools.

I'll be writing a series of articles about my students' experiences with the Altair 680b computer starting with their initial assembly of the system and working up through beginning programming.

My students' previous experience in complex electronic kit construction enabled them to assemble the Altair 680b and interface it to an ASCII keyboard with very few problems.

Students were divided into several groups to assemble the kit. Some worked on the main circuit board, others on the front panel and the remaining students worked on the power supply and quality control.

Continued

STUDENTS FIND ALTAIR 680B KIT EASY TO ASSEMBLE

Continued

However, as anyone who has assembled an Altair 680b will admit, it's difficult to collect the ideal tools for mounting components and completing the mechanical assembly of a microcomputer. The soldering iron is the most important tool. To assemble the Altair 680b kit, my students selected the Ungar "Princess" microline soldering iron with an 18 watt heat cartridge and the iron clad precision soldering nib. This tool is perfect for all delicate soldering operations, expecially IC sockets. The 18 watt heat unit won't damage components unless the iron is left on too long. One of my students came up with the idea of using a clothspin heatsink to prevent such accidental heat damage to ICs during the soldering process.

A major concern when assembling an Altair computer kit is the effects of static electricity on MOS-IC's and how this static can be eliminated. My students solved the problem by "zapping" each MOS device with a Zerostat gun. A Zerostat gun is a Piezo-electric gun that shoots out a stream of positive ions and has the capability of completely neutralizing any static charges on any item--MOS devices, plastic records or even clothing.

It's also important to be sure that there are no solder bridges across those IC solder connections. Edmund Scientific's catalog contains a number of magnifiers with magnifications up to over 12X--just the ticket to examine those tiny bridges at the IC socket solder points. My students used one of these magnifiers to check every solder connection on the display panel circuit board as well as on the main processor circuit board.

For experience in troubleshooting, I told my students to develop a simple clip-on DIP logic status tester. In my next article I'll discuss this project and the debugging process that my students used.

LOW COST GRAPHING HELPS MATH STUDENTS

By Art Armstrong

Reprinted from SCCS INTERFACE, January, 1977, pp. 23-25.

SCCS INTERFACE Editor's Note:

This article documents the graph drawing program which Art Armstrong uses in his math classes at Venice (California) High School. The user specifies a function and the location of the origin on the display, and the program plots it on a video monitor.

Several years ago, I installed a time sharing terminal in the public library in my community. Naturally, students from the high school began to come around, and I began to hear about this fantastic math teacher, Mr. Armstrong. Since then I've gotten to know Art Armstrong, and can understand his studens' enthusiasm--he is a dedicated, innovative teacher. Art has used time sharing and an HP calculator-plotter for years. He is the first teacher I know of to have used an Altair computer in the classroom and the first to have students assemble Altair computers in the classroom.

Art is more than an equipment freak. He is the sort of teacher whom students come back to visit years after graduation. His classroom is open and active and he is one of the few public school teachers I know who is around after school hours. Art has been very active in SCCS and is a member of the board of directors.

At one time or another, every math teacher and student has probably longed for a simple, affordable device that produces graphs of functions in response to mathematical statements. The promise of such a device has been apparent for years, but the actual hardware has been a long time in coming.

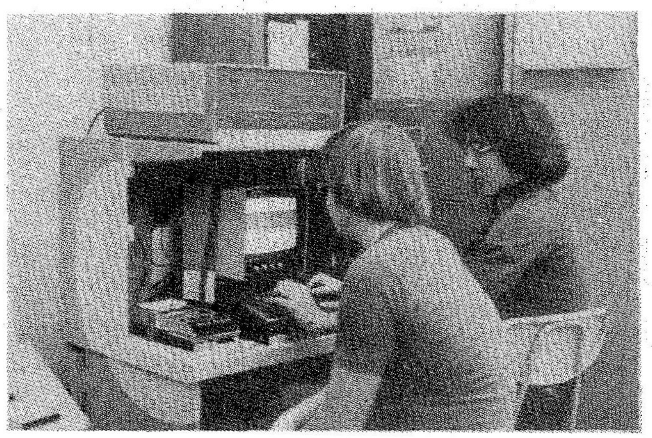
About 10 years ago, analog plotters were placed on the market by calculator manufacturers such as Monroe and Hewlett Packard. These plotters produced graphs by drawing short straight line segments approximating the curve. They were dramatic to watch and useful in application. But they were also expensive and required use of the rather cumbersome language of the driving programmable calculator.

Five years ago, Hewlett Packard and Wang produced plotters which were driven by BASIC-speaking calculators. These were easy to use and produced excellent graphs. But the prices ran close to $8,000(with calculator).

Recently, Tektronix came out with a CRT vector graphics unit which is fast, has a large capacity for data, speaks BASIC and has a hard copy option. Although the price, about $7,500, represents a leap in cost-effectiveness over previous plotters, it is still out of reach for most people.

By plugging the Polymorphics Video Display into a computer using the Altair bus, graphics are now affordable. Video display units not only allow graphics but also serve as standard alphanumeric input-output devices. Using a version of BASIC with a POKE statement in conjunction with a video display unit produces graphic output at a fairly fast rate.

From a graphics viewpoint, the only disadvantage of the video board is that the resolution is low. The Polymorphics board offers 128 horizontal and 48 vertical units in graphics mode. This is better than that offered by the Teletype, which has traditionally been used in math programs for graphing. But it lacks the accuracy and continuity of the



Two students using the graphing system.

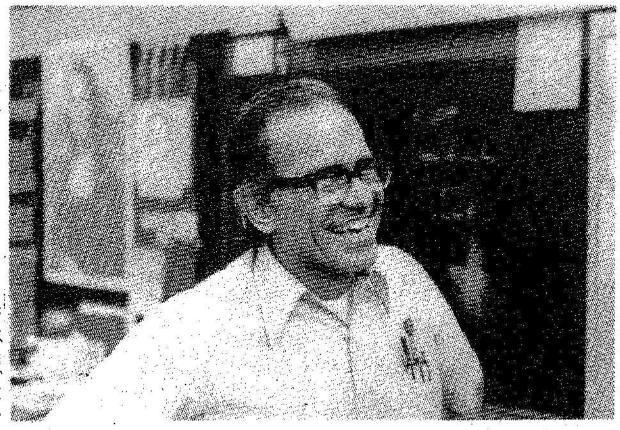


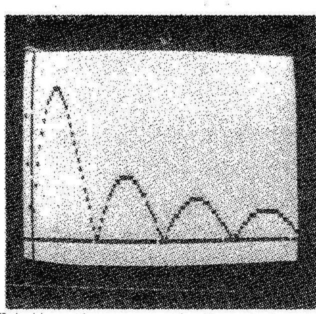
Photo by Larry Press

Tektronix units. However, the Polymorphics board is the most affordable and easy to use.

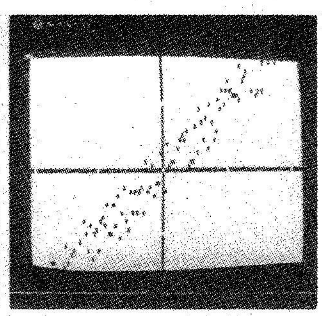
I used the program described on page   in a high school classroom to help students gain a better understanding of the relation between an equation and its graph. Even with the low resolution, students can learn much about the behavior of functions in a dramatic, dynamic and responsive manner. Questions which cannot be resolved with the unit, such as continuity, roots, intersections and asymptotes, can be left for other analysis wherever necessary.

This program will run in an 8K machine using MITS 8K BASIC. The listing is shown with a minimum number of REMs to make it easier to copy. Comments relate to line numbers in the listing.

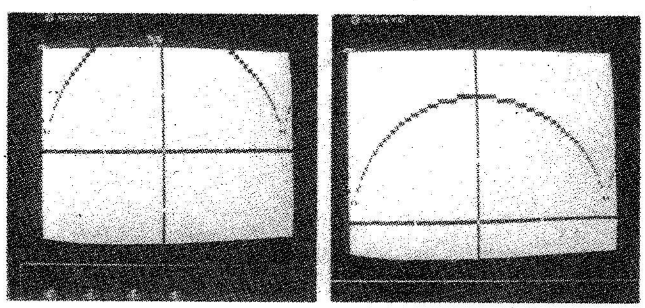
line 10	SA is the starting address in decimal of the polymorphics board. This should be changed to suit your installation.
line 15	RO is the row of the origin. The display rows are numbered from top to bottom, 1 to 16. Selecting one of these for RO will locate the X-axis on that row.
line 16	CO is the column of the origin. The display columns are numbered left to right, 1 to 64. Selecting one of these for CO will determine the position of the Y-axis.
lines 120-127	Constants are set as variables to save conversion time.
line 126	YC is the Y conversion constant, three units of display per unit of Y or about 16 units of Y from top to bottom.
line 140	Array M contains the graphics values to be used in plotting the function at line 1550.
line 200	This line clears the screen to all white. Plotting is black on white.
lines 490-530	This routine draws the X-axis. The POKE at line 520 marks the top third of the origin row as the X-axis.
lines 535-570	This routine draws the Y-axis. The POKE at line 560 marks the left half of the origin column as the Y-axis.
lines 595-750	This routine clears spots on the axes at five-unit intervals.
line 780	This line cleans up the origin.
lines 1000-1900	This is the routine that calculates, converts, and draws the function.
line 1000	GC is the display column from 1 to 64. This is divided into left and right half at line 1100.
line 1200	Here the value of X is calculated as a function of the graphics column number. C is the graphics column, 1 to 128. AC is the conversion for the Y-axis location, and W6 is the scaling factor for X, about 6 graphics columns per unit of X or 21 units of X, left to right. Changing W6 in proportion with YC (line 126) will change the scaling of the graph.
line 1250	This is the location for the expression of Y as a function of X. Any legitimate BASIC expression may be used.
line 1350	This converts the calculated Y value to a graphic row number, 1 to 48.
lines 1400-1401	These test the row value to see if it is off screen.
line 1450	This converts the graphics row (1 to 48) into a display row (1 to 16).
line 1500	IR is the internal row number (1 to 3) within the display row.
line 1550	Here the proper graphics value is selected from the M array depending on the position of the plotted point within the display row and column matrix.



Graph of a bouncing ball function.



A graph of y = x with uniform random noise added.



The origin may be placed at any point on the screen. Here a graph is moved down in order to fit.

Continued on page Eighteen

The ALTAIR™ ADVANTAGE

by **Barry J. Yarkon**
Vice President, PhotoSystems
Graphicomposition, Inc. (NYC)

A recent article described our Altair 8800 implementation in the Graphic Arts. In response to that article came phone calls and correspondence from such faraway places as the United Kingdom, Israel and Spain! Each was from people in small to medium sized businesses who wanted to know more about how we had learned "...to write custom text editing and file manipulation programs that give my company a competitive advantage in the typesetting marketplace."[1]

We have since discovered that the effect on our company is even more subtle than merely a competitive *production advantage*—the Altair has the potential of becoming a dynamite *marketing* and *sales* tool as well. This means profit dollars in addition to the daily production savings that efficient computer-assisted editing yields.

This article will present an actual situation in which the power of our microprocessor-based system got the job done and allowed us to save-the-day for one of our clients. We not only produced his job quickly and inexpensively, but we did so with almost no effort on the part of his staff—a fact he greatly appreciated. Providing a unique service which resulted in increased sales volume (for us) and making our client look good to *his* superiors, is a marketing advantage which profits all involved.

The Problem

Figure 1 shows a sample column from a booklist which we had previously typeset for our client (a large book publisher). It was originally input on paper tape keyboards, periodically updated and run through our phototypesetting machines. For the current revision we read these standing paper tapes into the Altair and onto floppy discs. Editing was then done via a display terminal, and the updated records were later sent on-line to the phototypesetters (*viz* Figure 1).

Graphically, each item consists of left and right dotted lines, a book number, title, author and price. There are over 4,000 items in this one publication.

In order to appreciate the problem that arose, you will need to examine the book numbering system that publishers use—the International Standard Book Number or ISBN. This is a worldwide convention, agreed to by almost all publishers, which sets forth a standard for producing unique identification numbers used on every published book, its dust cover, shipping cartons, catalogs, invoices and order forms. The ISBN Agency publishes a monograph[2] which, in brief, sets the format of Figure 2.

Crucial to the system is the *check digit*, which is derived by a mathematical procedure involving modulo 11 arithmetic. The check digit allows verification of ISBNs during inventory and order fulfillment. This is usually done in large computers by duplicating the standard arithmetic and then comparing the derived check digit with the one input as data. A difference between the two shows the whole ISBN entered is in error and recovery procedures are enacted.

As you can see in Figure 1, our client uses only the five digit portion of each book number (i.e., 20873) in his list. Now the problem: this publishing firm's own single largest client (a national retail bookstore chain) requested a version of the booklist with both their imprint *and* check digits!

Second color denotes 1977 publications

....	**20873**	**ABORTION Brautigan**	**p. 2.95**	....
....	22502	ACKROYD Feiffer (May)	8.95	....
....	20486	ADVENTURE OF BIRTH Bing	4.95	....
....	**21755**	**ADVENTUROUS CROCHETER Brock & Bodger**	**p. 4.95**	....
....	22242	ADVICE TO MY GRAND-DAUGHTER Hough	8.95	....
....	22785	AFTER CONVICTION Goldfarb & Singer (Mar.)	p. 6.95	....
....	21206	•cloth	19.95	....
....	21468	AFTER THE BALL Whitcomb	7.95	....
....	21530	AFTER THE PLANNERS Goodman	p. 3.45	....
....	20981	•cloth	9.25	....
....	**22062**	**AGAINST OUR WILL Brownmiller**	**10.95**	....
....	01401	AGONY MODERN MUSIC Pleasants	p. 2.95	....
....	**27065**	**AINSLIE'S COMP. GD. HARN. RACING**	**12.95**	....
....	**01469**	**AINSLIE'S COMP. GD. THRBD. RACING**	**14.95**	....

Figure 1: Original booklist format.

Second color denotes 1977 publications

....	**20873-X**	**ABORTION Brautigan**	**p. 2.95**	....
....	22502-2	ACKROYD Feiffer (May)	8.95	....
....	20486-6	ADVENTURE OF BIRTH Bing	4.95	....
....	**21755-0**	**ADVENTUROUS CROCHETER Brock & Bodger**	**p. 4.95**	....
....	22242-2	ADVICE TO MY GRAND-DAUGHTER Hough	8.95	....
....	22785-8	AFTER CONVICTION Goldfarb & Singer (Mar.)	p. 6.95	....
....	21206-0	•cloth	19.95	....
....	21468-3	AFTER THE BALL Whitcomb	7.95	....
....	21530-2	AFTER THE PLANNERS Goodman	p. 3.45	....
....	20981-7	•cloth	9.25	....
....	**22062-4**	**AGAINST OUR WILL Brownmiller**	**10.95**	....
....	01401-3	AGONY MODERN MUSIC Pleasants	p. 2.95	....
....	**27065-6**	**AINSLIE'S COMP. GD. HARN. RACING**	**12.95**	....
....	**01469-2**	**AINSLIE'S COMP. GD.**		

Figure 4: Revised format for booklist.

Group Identifier — Publisher's Number — Title Identifier — Check digit

W-XXX-YYYYY-Z

I.e., **0-671-20506-4**

Figure 2: ISBN description.

To comply with this request using conventional methods, our client would be forced to redo the typesetting from scratch. It would have involved a very tedious job of looking up thousands of check digits from an inventory computer printout; copying them by hand onto an old printed book list, sending this as manuscript to a typographer, proofreading the newly typeset galleys, etc. A week or more and several thousand dollars would have been involved.

The Altair Advantage

When consulted about this problem, we realized the solution could be more direct. After obtaining a copy of the ISBN Monograph, we found that Disc Extended BASIC already had a MOD operator. So, knowing the arithmetic algorithm and that the "0-671-" prefix was constant for all books on the list, within three hours we had written a small program to accomplish the job.

Figure 3 is a listing of that program dubbed "ISBNFIX". It opened the original booklist file-by file on diskette, extracted each record's five-digit book number, calculated the check digit and put the record back on diskette—number, hyphen and all. When these new files were sent to our phototypesetters, the resultant typeset galleys contained the requested format (see Figure 4). Elapsed time: one day. Accuracy: 100%.

Putting yourself in our client's place—to which typographer would you send your work?

[1] COMPUTER NOTES, Volume 2 Issue 6, November 1975.

[2] The ISBN System: Users' Manual, International Standard Book Numbering Agency, Berlin, 1975.

```
5 REM: AUTO EXTRACT & CALC.
       ISBN CHECKDIGITS FR
       BOOKLIST FILES.
      Author: BJ Yarkon
100 NULLO:CLEAR1024:PRINTCHR$(12)
110 INPUT"* FILE NAME; DRIVE";Y$,X
115 INPUT"* FROM RECORD NO.";Z
120 OPEN"R",1,Y$,X
130 FIELD#1,128ASS$
140 GET#1,Z:D$=""
150 IF LOF(1)=LOC(1)-1 THEN 490'STOP
155 REM:
         SEARCH FOR VALID 5-DIGIT NO.
         FIELDS: *1<NUMBER>*2 OR
                 *f<NUMBER>*2


160 FOR I%=1TO LEN(S$)
170  IF MID$(S$,I%,2)<>"*1" AND
      MID$(S$,I%,2)<>"*f"GOTO200'NEXT
180  D$= MID$(S$,I%+2,5):GOTO300'FOUND
200  NEXTI%'STILL LOOKING!
210 GOTO500'not found in record!
295 REM:
         OK, CALC. ISBN CHECK DIGIT


300 R=117:K=6:N=0
310 FOR J%=1TO5
320  Q=VAL(MID$(D$,J%,1))*K
330  K=K-1:R=R+Q
340 NEXTJ%'EXITS WITH SUBTOTAL CNT
350 IF (R+N)MOD11=0THEN 370'N=CK
360 N=N+1:GOTO350'LOOP
370 IF N=>0 AND N<=10GOTO390
380 PRINT"*ERROR, N="N:CLOSE:STOP
390 IF N=10 THEN N$="X":GOTO410
400 N$=STR$(N)
410 A=I%+6:B=123-I%
412 REM:
         REASSEMBLE NEW STRING AND
         REPLACE IN FILE

415 NW$=LEFT$(S$,A)+"-"+RIGHT$(N$,1)+
    RIGHT$(S$,B-1)
420 LSET S$=NW$
430 PUT#1,Z
440 Z=Z+1: GOTO130' OK DO NXT RECORD!
490 CLOSE:END
500 OL$=S$
505 LSET S$=OL$'NO ISBN FOUND IN RECORD
510 PUT#1,Z
520 Z=Z+1: GOTO130' DO NXT RECORD!
```

Figure 3: Program listing, "ISBNFIX".

LOW COST GRAPHING HELPS MATH STUDENTS

Continued

line 1600	AD is the address for the POKE in line 1800.
line 1650	The selected graphics value is ORed with the present value of the display element to maintain the display at places where the graph crosses the axes.

```
1 REM POLY-PLOT 12/1Ø/76
2 REMBY ART ARMSTRONG
3 REM COPYRIGHT SCCS INTERFACE
1Ø SA=31744:REM SA IS THE STARTING ADDRESS OF THE POLYMORPHIC BOARD.
15 RO=9:REM ORIGIN ROW
16 CO=32:REM ORIGIN COL.
2Ø AR=3*RO-2
21 AC=2*CO-1
1ØØ DIM M(3,2)
11Ø P=SA-65
115 AX=B+576:AY=B+32
12Ø WN=1:TW=2:TH=3:S4=64
122 TY=2Ø:F6=56:F8=48:F5=15:T4=24
123 T6=36
126 YC=2.99
127 ZR=Ø:P5=.5:F4=4:SX=16:FV=5
13Ø W3=1/3:W6=1/6
135 BG=Ø
14Ø FORX=1 TO 3: FORY=1 TO 2:READYM(X,Y):NEXTY,X
15Ø DATA32,4,16,2,8,1
16Ø C=32
19Ø REM  CLEAR SCREEN
2ØØ FORX=B+65TOB+1Ø88:POKEX,BG:NEXT
49Ø REM DRAW X-AXIS.
5ØØ FORX=1 TO S4
51Ø AD=S4*RO+X+B
52Ø POKEAD,T6
53Ø NEXT
535 REM  DRAW Y-AXIS.
540 FORY=ZR TO F5
55Ø AD=CO+S4*Y+SA-1
56Ø POKEAD,F6
57Ø NEXT
59Ø REM  ADD 5 UNIT TICS ON X-AXIS.
6ØØ L=CO-F5*INT(CO/F5)
61ØR=CO+F5*INT((S4-CO)/F5)
62Ø FORX=L TO RSTEPF5
63Ø AD=S4*RO+X+B
64Ø POKEAD,F4
65Ø NEXT
69Ø REM ADD 5-UNIT TICS ON Y-AXIS.
7ØØ T=RO-FV*INT(BO/FV)
71Ø BO=RO+FV*INT((SX-RO)/FV)
72Ø FORY=TTOBOSTEPFV
73Ø AD=CO+S4*Y+B
74Ø POKEAD,T4
75Ø NEXT
78Ø AD=CO+S4*BO+B:POKEAD,28
79Ø REM  CLEAR ORIGIN.
99Ø REM  START SCAN ON X
1ØØØ FORGC=1TO64
1Ø5Ø C=TW*(GC-WN)
11ØØ FORIC=WNTOTW
115Ø C=C+WN
12ØØ X=(C-AC)*W6
124Ø REM
125Ø Y=X:REM  EXPRESSION FOR Y GOES HERE
126Ø REM
135Ø R=AR-SGN(Y)*INT(APS(YC*Y)+P5)
14ØØ I FR>F8THEN185Ø
14Ø1 I FR<WN THEN 185Ø
145Ø GR=INT((R+TW)*W3)
15ØØ IR=R-TH*INT((R-WN)*W3)
155Ø IM=M(IR,IC)
16ØØ AD=B+S4*GR+GC
165Ø P=PEEK(AD)ORIM
18ØØ POKEAD,P
185Ø NEXT
19ØØ NEXT
```


HAM ON THE SIDE

By David Le Jeune

In a previous article I described MAILBOX, an amateur radio message store and forward system. Although MAILBOX was a "state of the art" application of microcomputer technology to amateur radio, several features made it impractical for the average ham to use. It was a disk oriented system that required 24K of system RAM and extensive patches to Altair Disk BASIC for input/output routines.

ARCOS, the Amateur Radio Computer Operating System, is a computer system that handles many of the details of operating an amateur RTTY station. It does have the store and forward feature, but it can be used with a 12K, non-disk operating system. It can be used with Altair 8K BASIC and requires no patches. The only requirement is a serial I/O port strapped for 45.45 baud (60 WPM), 5 level code (Baudot). ASCII to Baudot and Baudot to ASCII is done entirely in Altair BASIC.

The following design criteria were established in developing ACROS:

a. No patches to Altair BASIC
b. Software ASCII/Baudot/ASCII conversion.
c. Software generated CW (morse)identification
d. Error correction
e. Automatic insertion of end of line (EOL) sequence
f. 10 minute timer
g. Automatic answer-back (acknowledgement of messages)

ARCOS meets these as well as other goals. Although no attempt has been made to "tighten up" the code, the program will run in 12K, using Altair 8K BASIC. However, Altair Extended BASIC is recommended, since the ability to use integer variables, especially for loops, improves keyboard response time. The requirement that it run with no patches and that the CW identification and code conversion be done entirely in software caused some difficulties and lots of head-scratching. But flexibility afforded by the Altair BASIC PEEK and POKE and OUT functions and commands greatly eased this problem. Without them, these two design goals could not have been met.

The only other problem encountered was in the attempt to get data from the console keyboard using the INP function. After hours of trial and error debugging, I discovered that in order to do this reliably, the control C abort feature of Altair BASIC must be disabled. This is done by making the control C feature respond to an unused port. (Actually, this could be considered a patch, but since it is done with a POKE command in the initialization module of the program, it is transparent to the user.)

The program can be divided into three modules, an initialization module (lines 1-190), a receive module (lines 200-300) and a transmit module (lines 490-5200), which I will discuss in next month's C.N.

Initialization Module

This module sets up several ASCII string and numerical arrays. Lines 10-73 set up the Baudot to ASCII conversion table A$(i). Lines 80-130 set up the ASCII to Baudot conversion table B(i). Line 8 establishes the CW speed, and line 9 determines the "dot up," "dot down" and "dash" ratios. This timing permits the CW message to be "printed" on a 60 WPM Baudot printer as a series of dots. CW identification is required by the Federal Communications Commission every 10 minutes or after each transmission. Lines 140-190 set up the CW message string array I$(i). In the array a "-" represents a dash, a "." represents a dot, and a "/" separates characters.

Receive Module

The INP function is used to alternately check the ASCII status port (port 18) and the Baudot status port (port 20). Whenever an ASCII character is ready, it is retrieved from the ASCII data port (port 19) and checked to see if it is a "B," "T" or "DEL." A "B" or "T" causes the program to branch to the transmit module. The "T" causes a CW identification message to precede the RTTY transmission. A "DEL" causes the program to revert to BASIC command level. This is necessary because the control C abort feature has been disabled. Any ASCII character other than "B," "T" or "DEL" is ignored.

If a Baudot character is ready, the variable J is assigned the value of the data at the Baudot data port (port 21). If it is either a blank (J=0) or a carriage return (CR, J=8), it is ignored. A line (LF) character (J=2) causes the printer to turn up a new line. A space causes the shift condition flag to be reset (A=0). The shift condition flag "remembers" whether the last Baudot shift character was a letter or a figure shift. Down shift (to letters) on space is a common technique used in amateur radio teletype (RTTY). The receipt of a "letters" or "figures" character causes the variable A to be set to 0 or 32 respectively.

```
LOAD "ARCOS"

OK
LIST

1 REM ARCOS, AN AMATEUR RADIO COMPUTER OPERATING SYSTEM.  4 AUG 76
2 DEFINTA-Z
3 S$="      "
4 M$(0)="DE DAVE K5WNV/7 FORT HUACHUCA, ARIZONA"
5 DIMA$(63),B(63),I$(63),L(63)
6 M$(1)="DE DAVE K5WNV/7 FORT HUACHUCA, AZ"
8 CW=55
9 DD=CW*.66:DU=CW*.88:DA=CW*2.09
10 FORI=0TO63:READA$(I):NEXT
40 DATANUL,E,LF,A,SP,S,I,U,CR,D,R,J,N,F,C,K,T,Z,L,W
50 DATAH,Y,P,Q,O,B,G,FIG,M,X,V,LET,NUL,3,LF,-,SP,'
60 DATA8,7,CR,ACK,4,',COMMA,!,COLON,(,5,+,),2
70 DATA#,6,0,1,9,?,&,FIG,.,/,=,LET
71 A$(44)=","
72 A$(46)=":"
73 A$(49)=CHR$(34)
80 FORI=0TO63:READB(I):NEXTI
90 DATA4,45,0,52,0,0,58,37,47,50,0,49,44,35,60,61,54,55,51
100 DATA33,42,48,53,39,38,56,46,0,0,62,0,57,0,3,25,14,9,1
110 DATA13,26,20,6,11,15,18,28,12,24,22,23,10,5,16,7,30,19
120 DATA29,21,17,31,0,27,0,0
130 B(2)=49:B(4)=41:B(27)=62
140 FORK=0TO49
150 READI$(K)
160 NEXTK
170 DATA-,.,-,/,.,.,.,.,/,.,-,-,/,-,.,/,.,.,.,-,/,/,/,/
180 DATA-,.,.,-,.,/,-,-,.,.,.,/,/,/,/,/,/,/,
190 DATA/,/,/,/,/,/,/
200 IF(INP(0)AND1)=0THEN209
201 IF(INP(18)AND1)=0THEN200ELSEJ=(INP(19)AND127):IFJ=84THEN500
202 IFJ=66THEN510
203 IFJ=127THENSTOP
204 IFJ=67THEN1000
205 IFJ=87THEN1200
208 GOTO200
209 J=(INP(1)AND127)
210 IFJ=4 THEN PRINT" ";:A=0:GOTO200
215 IFJ=0THEN200
220 IFJ=8THEN200
230 IFJ=2THENPRINT:GOTO200
240 IFJ=31THENA=0:GOTO200
250 IFJ=27THENA=32:GOTO200
255 J=J+A
260 IFA$(J)="COMMA" THENPRINT",";:GOTO200
270 PRINTA$(J);
280 S$=RIGHT$(S$,5)+A$(J)
290 IFS$="WNVWRU"THEN1500
291 IFRIGHT$(S$,2)="BK"THEN490
300 GOTO200
490 IF(INP(255)AND64)=0THEN200ELSE510
500 OUT0,1:GOSUB1900:GOSUB1920
510 OUT0,1
520 IF(INP(18)AND1)<>1THEN4000
525 J=(INP(19)AND127)
526 IFJ>31THEN570
527 IFJ=7THEN5000
530 IFJ<>13THENPRINTCHR$(J);ELSEGOSUB900:PRINT:L1=0:GOTO520
540 IFJ=18THEN800
550 IFJ=1THENGOTO5200
551 IFJ=6THEN1000
552 IFJ=4THEN1100
553 IFJ=5THEN1200
554 IFJ=14THENPRINT:INPUT"CW SPEED";CW:GOSUB2000:GOTO520
555 IFJ=26THENM$=M$(1):GOSUB1800:GOSUB900:PRINT:GOTO520
560 IFJ=2THEN830
570 IFJ>95THEN3900
571 PRINTCHR$(J);
580 J=B(J-32)
590 IFJ>31THEN620
600 IFA=0THEN650
610 A=0:L(P)=31:P=P+1:IFP=64THENP=0
615 GOTO650
620 IFA=1THEN650
630 A=1:L(P)=27:P=P+1:IFP=64THENP=0
635 GOTO650
650 IFJ=2THEN3910
660 L(P)=J:P=P+1:IFP>63THENP=0
670 L1=L1+1:IFL1>63THENGOSUB900:PRINT:L1=0
671 IF(L1>54)AND(J=4)THENGOSUB900:PRINT:L1=0
680 GOTO3920
800 GOSUB1990
810 OUT0,0
820 GOTO200
830 OUT0,0
840 GOTO200
900 FORM=0TO2
905 L(P)=8:P=P+1:IFP=64THENP=0
910 NEXTM
920 L(P)=2:P=P+1:IFP=64THENP=0
930 L(P)=31:P=P+1:IFP=64THENP=0
940 RETURN
```

Continued

HAM ON THE SIDE

Continued

If the Baudot character received is not of those identified in the preceding paragraph, J is added to A, and then the ASCII equivalent A$(J+A) is retrieved and printed. For example, if J=L, J could represent either a Baudot "E" or "3," depending on whether the last shift character received was a "letters" or "figures" shift. Assuming it was a "figures," then

A$(J+A) = A$(1+32) = A$(33) = "3"

After the character (line 270) is printed, it is appended to the string S$ representing the preceding 6 Baudot characters received. The oldest character in the string (the first one) is then dropped. If S$ = "WNVWRU," this is interpreted as a request for acknowledgement. So the program branches to line 1500 to transmit the acknowledgement message. If the last two characters received are the characters "BK," this is interpreted as a possible fast break command, so the program branches to line 490 to check the setting of the sense switches. If sense switch 13 is up, the system branches to the transmit module. Thus, the setting of sense switch 13 controls the capability of the sending station to turn the receiving station transmitter on by sending the sequence "BK."

Other sequences can be added to the system. But with each new sequence, more time is required for decoding. There is also a limit to the number of sequences that can be added. If this limit is surpassed, characters will be lost while the system is decoding sequences.

```
1000 OUTO,1:GOSUB2000:M$="CQ CQ CQ CQ DE DAVE K5WNV/7 FORT HUACHUCA, AZ"
1010 FORK=0TO3
1020 GOSUB1800
1025 GOSUB1900
1026 PRINT
1030 NEXTK
1040 M$="K K K K K K  CW ID FOLLOWS"
1050 GOSUB1800
1060 GOSUB2000
1070 OUTO,0
1080 GOTO200
1100 M$=M$(0):GOSUB1800
1110 GOSUB1900
1130 GOTO520
1200 A=0:INPUTM$:OUTO,1:J=31:FORI=1TO30:GOSUB5100:NEXTI
1210 GOSUB1800
1220 J=31:GOSUB5100:J=0:GOSUB5100:J=25:GOSUB5100
1230 OUTO,0:GOTO200
1500 M$=M$(0)
1505 OUTO,1
1510 GOSUB1920
1520 GOSUB1800
1530 GOSUB1900
1540 GOSUB1990
1545 OUTO,0
1560 GOTO200
1800 FORI=1 TO LEN(M$)
1810 J=B(ASC(MID$(M$,I,1))-32)
1811 IFJ>31THEN1816
1812 IFA=0THEN1820
1813 A=0:WAITO,128,128:OUT1,31
1814 GOTO1820
1816 IFA=1THEN1820
1817 A=1:WAITO,128,128:OUT1,27
1820 WAITO,128,128:OUT1,J
1825 PRINTMID$(M$,I,1);
1830 NEXTI
1840 RETURN
1900 J=8:GOSUB5100:J=8:GOSUB5100:J=2:GOSUB5100:J=31:GOSUB5100:PRINT
1910 RETURN
1920 J=31:FORI=1TO15:GOSUB5100:NEXTI:RETURN
1990 M$="NNNNN":GOSUB1800:GOSUB1900
2000 WAITO,128,128:OUT1,27
2005 FORI=0TO300:NEXTI
2010 FORU=0TO40
2040 IFI$(U)="-"THENI=DAELSEIFI$(U)="."THENI=DDELSEJ=DA:GOTO2200
2120 OUTO,9
2130 FORK=0TOI
2140 NEXTK
2200 OUTO,1
2210 FORK=0TOJ
2220 NEXTK
2230 J=DU
2250 NEXTU
2255 FORI=0TO100:NEXTI
2260 GOSUB1900
2300 RETURN
3900 T1=T1+1:GOTO4000
3910 T1=T1+2:GOTO4000
3920 T1=T1+3
4000 IF(INP(0)AND128)<>0THEN520
4001 T1=T1+1:IFT1<T2THEN520
4002 T1=0
4010 IFO=PTHENOUT1,31:A=0:GOTO520
4020 OUT1,L(O):O=O+1:IFO=64THENO=0
4030 GOTO520
5000 INPUTT2:GOTO4000
5100 WAITO,128,128:OUT1,J:RETURN
5200 INPUTM$(1):GOTO520
```

GLITCHES

"Window" Program Isolates System Faults

By Bruce Fowler

Troubleshooting Input/Output system faults is complicated by the fact that problems in the terminal or I/O board can appear as trouble anywhere else in the system. The reverse is also true. CPU or memory problems can sometimes appear as I/O errors.

To help isolate I/O problems from other system faults, a "window" program is often used. This is a program that accepts input from an I/O port and displays it on the computer's front panel lights. By typing characters on the terminal and observing the bit patterns on the front panel lights, the operation of the I/O board may be observed.

The MITS serial I/O boards have either a 6850 ACIA (2SIO) or a 2502 UART (SIO A, B or C, ACR) to perform the parallel to serial and serial to parallel conversions. These circuits also perform parity and framing error checking as well as other housekeeping chores. When a complete serial character is received without error, it is stored in the data register. On signal from the CPU, the character is output from the data register on the parallel data lines DI0 through DI7.

Window Programs

For I/O boards with UART, MITS repair technicians use the simple window program in Table 1. To use this program, actuate the Single Step switch twice from the beginning of the program. (Note: don't use the M1 single step switch on the 8800b.) Whatever data is in the UART data register is now displayed on the Data lights. Pressing any character key on a terminal connected to the I/O board causes the ASCII code of the character to be displayed on the data lights. For example, the character 'A' is displayed as 01000001 or 101 octal. A complete list of ASCII character codes can be found in the Altair BASIC manual, Appendix A.

This same program can be used to display the changes in the status bits of the UART's control register by substituting the address of the control channel for the data channel address in location 001.

For boards using the ACIA, the program in Table 1 can be used to display the status bits. But changes must be made to display the contents of the data register. These changes are necessary because an ACIA must receive a character, convert it to parallel and then receive a Read signal on pin 13 before that character can be output. So a new input instruction must be executed after every new character is received in order to see that character displayed.

An example of such a modified window program is shown in Table 2. To use it, single step 10 times from the beginning, type a terminal key and then single step two more times. Remember, don't use the M1 single step option on the Altair 8800b. The ASCII code of the character should be displayed on the data lights. To display other characters, single step four times, type a key and single step twice more for each character.

This program may be run (instead of single stepped) on an Altair 8800b to display the ASCII codes on the data lights. To do this, a jumper must be installed on the interface board from JA to JB. The 8212 latch on the interface board retains the last byte input from the I/O board while the computer runs. To do this on an Altair 8800a, the program in Table 3 must be used.

To check the serial input to the UART or ACIA, monitor the input signal at the serial input pin of the chip. This is pin 20 on the 2502 UART and pin 2 on the 6850 ACIA. Since the start bit is active low in both cases, the receive pin goes from high to low whenever a character is received. For a description of the internal workings of the UART, see p. 43 of the June 1975 issue of POPULAR ELECTRONICS.

Table 1

Location	Octal Code	
000	333	INPUT from data change address

Table 2

Location	Octal	Code
000	076	Master reset the
001	003	ACIA by sending 003
002	323	to control channel.
003	020	
004	076	Supply the ACIA with
005	021	format information,
006	323	i.e. number of bits
007	020	per character
010	333	Input the character
011	021	from data channel.
012	303	Loop back to input
013	101	more characters.
014	000	

Table 3

Location	Octal Code	
000	006	Set up counter
001	001	in B register.
002	076	Master reset ACIA
003	003	by sending 003
004	323	to control register
005	020	
006	076	Supply ACIA with
007	021	serial format
010	323	information
011	020	
012	333	Input character
013	021	from data address.
014	127	
015	032	This will display
016	032	character on address lights.
017	032	
020	032	
021	032	
022	004	Increment counter
023	302	If counter ≠ 0,
024	015	keep displaying
025	000	character.
026	303	Loop back for
027	010	next character.
030	000	

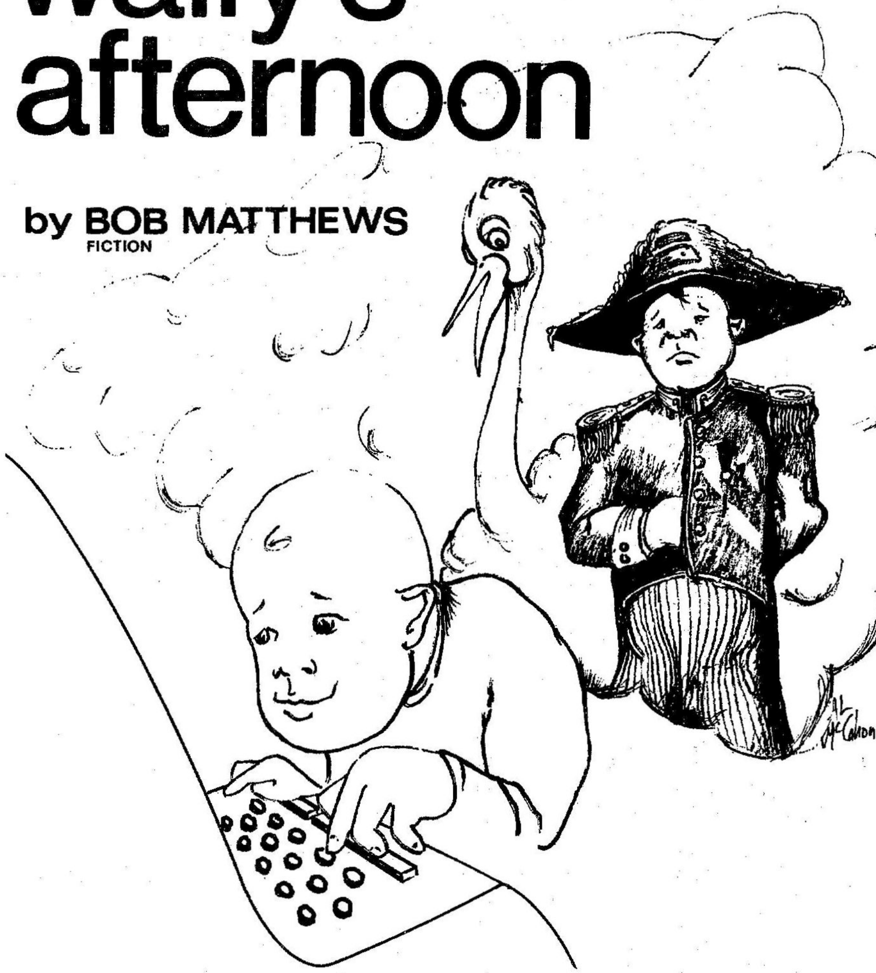
CORRECTION

In "Altair Disk Drive Alignment Permits Precise Operation" (see April C.N., pp. 5-7, 12 and 13), Figure 2 should be labelled "head properly aligned;" Figure 3, "head improperly aligned;" and Figure 4, "index sensor alignment oscilloscope pattern."

CLASSIFIED ADS

wally's afternoon

by BOB MATTHEWS
FICTION



"I could get worried about this if I could figure it out." He held the letter closer to his eyes, as if the crisp type would yield some information he hadn't seen in the past three days.

"Dear Mr. Walters," it said. "I have been advised that there is no record of your son Wally having attended school for the past year. Let me remind you that state law requires all children up to the age of 16 to attend an accredited school and that failure to do so is a third degree misdemeanor..." The letter was signed by a Mr. Blackwalder, lawyer for the public schools.

"There's got to be a mistake," Bob Walters thought. A level-headed man, 30-year-old Bob rarely got rattled about anything. But this letter made him feel strangely nervous. "We've sent Wally off to third grade everyday for a year. He loves school and his teacher. He's always coming home with stories about dinosaurs, faraway lands and all the other things eight-year-olds learn about. They must have mixed up the records," he concluded. Angrily tossing the letter on the kitchen table, he walked into the living room and slouched wearily into his dingey brown reclining chair. "How can they expect me to believe some lawyer and not my own experience every morning for the past year?" he mumbled to himself. Glancing over to the cluttered corner of the living room that constituted his workshop, he saw Wally silently staring at the blinking lights on the front panel of the old computer. His bushy red hair blocked Bob's view of the terminal. "It must be a computer foul-up," he thought. "The system slipped a disk or swallowed a byte or something."

That kind of error might be expected from Alvin, his old microcomputer but not from the International Data Controls 9000 or whatever it is that rules the data processing center at the public school's main office. Bob's mind drifted back eight years to when he was a computer novice and bought Alvin.

Alvin was one of the first home microcomputers. When Bob first bought Alvin, puzzled friends asked, "But what are you going to use it for?" Bob's response of "almost anything" wasn't too popular then. But Alvin and all the other home microcomputers had since proved that answer to be almost literally true.

Bob even made the newspapers with one of his "practical applications" seven years ago when Wally was just a baby. He had hooked up his computer to a small TV camera that watched over Wally's crib. By putting reflective spots on the corners of the crib blanket, the computer could determine(theoretically) whether the baby became uncovered during the night. Bob fashioned a clumsy mechanical arm out of some aluminum TV antenna tubing and some fishing line with a couple of surplus motors and potentiometers for servos. When the contraption was working, it could pull the blanket back up around Wally's chin if he kicked it off during the night.

But most of the time the thing didn't work. Once the arm got stuck and the linkages started to vibrate so loudly that they woke Wally up at 2 AM. Bob's wife, Nancy, usually a very quiet redhead, flew out of bed when she heard Wally's crying and the clanking noise of the monster. When she saw the metal arm vibrating precariously over Wally's bed, she became hysterical and flung a half-full baby bottle at it and bent some of its tubing. Years later, the servos and a couple of linkages still cluttered Bob's workbench.

During those early home computer days, Nancy's Scotch heritage, which was evident in a

practicality that bordered on austerity, led her to believe that personal computers were merely a fad comparable to goldfish swallowing. But in time she gradually realized that microcomputers could be very useful.

Alvin, and later a more modern computer(as yet unnamed), soon became almost indispensible around the house. Bob wrote routines to access recipes from a large file according to nutritional content, nationality, cost and available ingredients. The computer also kept a running file on payments for medicine and doctors bills, savings interest, church offerings and state and local taxes--all those tax deductible expenses that Bob and Nancy used to forget. The first three years of using the computerized file had saved them enough money to pay for a new memory board to expand the system. (It also paid for new carpeting in the bedroom and new insulation in the attic. Nancy saw to that.)

Before long, Nancy was using the computer almost as much as her husband. She started a computerized social policy discussion group with a number of other people around the country. Each participant used a computer to store his or her comments, edit them and send them by telephone to the other participants. This discussion could conceivably go on continuously for years without the participants ever meeting one another.

Nancy and Bob weren't the only members of the family who learned to rely upon the computer. The ill-fated baby-minding monster was only the beginning of Wally's introduction to the computer. When Bob constructed a new system, Wally had Alvin to himself. As soon as Wally was tall enough to reach the keyboard, he started using the games and educational programs Bob had written for him. By the time Wally started school, he could write his own programs to work arithmetic problems and draw pictures. But most of the time Wally simply enjoyed playing with Alvin as if the computer was simply another friend. Even Nancy had to admit that Alvin was one of the best babysitters Wally ever had.

Arousing himself from his daydream, Bob noticed that Wally was still hunched over the terminal, sitting with one leg under him on a telephone book in a kitchen chair. Except for the muffled pockle of the keyboard and huffing of the cooling fans, the only sound was Wally's soft giggles.

The slam of the back door broke the silence. It was Nancy. She had been picking flowers in the garden and carried a large bunch of chrysanthemums. "Isn't it about time to go? You have that meeting with what's-his-name, the lawyer," she said.

"Blackwalder," Bob said glumly as he struggled to return the chair to an upright position. "I know. We're going."

"Don't worry," she said as she laid the flowers on the kitchen table. As usual, she knew exactly what was bothering her husband. "It's all a big mistake."

"Yeah. A mistake," Bob said, still trying to ignore the doubt in his mind. "Come on Wally. It's time to go. You'd better say so-long to Alvin for now."

"O.K. Dad. Just a minute." That usually meant he'd be ready in an hour or two.

Bob headed for the door, fumbling in his pockets for the car keys. "Not in a minute," he said gruffly. "Now! Ah, here they are." The keys were on the end table where he had left them the night before. Bob noted with concern that he became absent-minded before he had to do anything unpleasant.

"Now let's get cracking, Wally." As Bob whirled around to prod Wally into action, he almost knocked him over. Wally already had his coat on and had even brushed his hair.

"Let's go, Dad," he said cheerfully.

Bob folded the letter carefully and then stuffed it in his left coat pocket as he walked out to the car. Although the leaves were already falling, the temperature was quite mild. Yet, the wind had a wintery chill to it. Bob tried to tell himself that he was always moody in the fall. He certainly wasn't looking forward to this meeting. "But, if Attorney Lawrence Blackwalder summons you to a meeting, you go," he muttered to himself as he nudged a yellow leaf off the sidewalk with his toe.

As they drove the short distance to school, Bob's anger grew. This shouldn't have come this far, he thought. Wally's teacher should have been able to clear up the whole matter.

Bob remembered how angry he had been during that meeting with Miss Sullivan, the third grade teacher. "Has Wally been in school or hasn't he?" he had finally demanded after a half-hour of pointless pleasantries. "Will you give me a straight answer or not?"

"You'll have to talk to Mr. Blackwalder," she said nervously. "I'm...I'm not allowed to discuss it."

As he downshifted to round a corner, Bob resolved to apologize to Miss Sullivan for the tirade he had launched about "bureaucratic underlings" who hadn't the brains to "think two thoughts" without their superior's approval. After all, this whole thing wasn't her fault. But Bob was still exasperated with the bureaucracy at the public schools. He was also a little frightened by what he might find out at this meeting with Blackwalder.

Bob was certain that Wally had been to school every day--except for the flu epidemic when virtually no one went. Blackwalder's threat about "third-degree misdemeanors" seemed to be just bluster. But why had the records suddenly disappeared, and why was the legal department of the school system so interested in all of this?

"Are you coming, Dad?" Wally's voice interrupted Bob's thoughts. The car, he noted, was already parked in the school's lot. "I must have been on automatic pilot," he mused.

They walked into the Administration Building (better known to everyone in the school system as Central Office, an ominous term that seemed particularly appropriate on this occasion) and asked the receptionist for Mr. Blackwalder's office.

"Follow me," she said in an officious tone as she lead them down a long, dark hallway.

The building looked just like any other government building. The walls of the hallway were painted a dull, two-tone green. The offices were marked by little black plastic signs with white letters that hung from wrought iron brackets. The receptionist stopped at one of them that said, "Legal Department, Mr. L. Blackwalder." With a forced smile, she said, "Go right in."

Blackwalder was a tall, brown-haired, middle-aged man. He was dressed immaculately in a blue pin-striped suit. He smiled nervously and carelessly shook hands with both Bob and Wally. Bob noticed that the lawyer's expensive clothes couldn't hide the beginnings of a paunch. "There's less to this guy than meets the eye," Bob thought mischievously.

"I've been expecting you," Blackwalder said, his smile turning into a sneer.

"Darn right you have," Bob thought. "You're the one who called this silly meeting." He and Wally sat down in two shiny, creaky leather chairs directly in front of Blackwalder's large desk.

Trying to sound as calm as possible, Bob said, "I don't believe that I know why you've asked us here." Two can play at this game, he thought.

Blackwalder gave Bob an exasperated look. But his smile was less confident. "Now Mr. Walters..."

"I think you had better explain to Wally and me what this is all about," Bob interrupted, "or you'll have to take it up with my lawyer!"

Continued

Bob amazed himself with his firmness. He didn't even have a lawyer.

Blackwalder's smile faded suddenly and was replaced by a look of severity that startled Bob.

"Do you mind if my secretary takes notes on our conversation?"

Before Bob could object, Blackwalder already had the phone receiver in his hand and was reaching for the intercom button. The noise that came from the receiver shattered the silence of the office. It was a much louder version of the annoying sound the telephone company uses when a phone is left off the hook. Bob glanced nervously over at Wally, who was looking at Blackwalder with complete detachment.

Blackwalder dropped the receiver as if it was infected. He stared angrily at it for a second and then quickly reached for it again. He punched another button and then lifted the receiver. The buzz split the air like a shot. He slammed down the receiver and started to get up from his chair. Then he noticed Wally, who was calmly studying the sole of his left shoe. Blackwalder sat down again, slowly and deliberately, and glared at Wally.

"You're doing this to me," he said in a desperate tone. "First the records and now this." Wally merely stared quizzically at him.

"Now Mr. Blackwalder, my son..." Bob began.

"I should say that this is all YOUR fault!" Blackwalder jerked his head to face Bob with vehemence. Yanking open his desk drawer, he pulled out a file folder and threw it on top of the desk. The newspaper article about the baby-minding monster was on top.

"I should have known it was you. This is all the proof I need," he said distastefully. "You've been behind this all along...you and your computers and whatever else. Where are those records?" he demanded.

"I'm afraid that I don't know what you're..." Bob faltered, genuinely puzzled and a little frightened at the outburst.

Blackwalder's eyes bored into Bob. "You knew all along that those records were missing," he said, his voice getting louder. "Not just your son's but all his little friends, too. Even his teacher's personnel records have been altered. She got two paychecks that were bigger than the superintendent's before we caught the mistake. Some mistake!" he snorted.

"Now just a darn minute," Bob shouted, jumping to his feet and leaning over Blackwalder's desk. Peering down at him, he said angrily, "You know I don't have access to your files. Sure, I've got a computer at home. Two of them, in fact. But we use them for recipes and income tax--NOT for stealing school files!" Blackwalder just sat there, amazed, staring at Bob. "What kind of idiot do you take me for? If you knew anything about your own files and computers, you wouldn't be able to fling accusations around like that. You can talk to my lawyer. We're getting out of here. Come on, Wally."

Bob turned to leave. But Blackwalder blurted out, "No! Ah...Wait a minute." He looked meekly at Bob and Wally. "I'm sorry for this...um...misunderstanding. I believe you if you say that neither you nor your son had anything to do with this. But I need your help. I've got to get to the bottom of this mess. Please, sit down. Explain to me why you couldn't have done this." He smiled uncertainly and motioned Bob and Wally to sit down.

So Bob explained in detail--as much as his limited knowledge of computer-based information storage and retrieval systems would permit--how a potential records thief would need a thorough knowledge of the inner workings of the school system to pull off a caper like stealing student files. The thief would also need a detailed knowledge of the school system's computers, since protection codes vary from machine to machine. Bob finished by saying, just for effect, that he had often wished that he had the information it would take for such a feat. Unfortunately, he added, since he couldn't think like a computer, the microcomputers he used were mystifying enough.

"You mean that you can't tell me how this was done?" Blackwalder asked, dumbfounded.

"No. In fact, I doubt it was 'done.' My guess is that it's all a fantastic coincidence."

Before Blackwalder could ask another question, the phone rang. "Ah, it's working again," he said with relief as he picked up the phone. As he listened, his eyes widened and his nostrils flared. "I see. Thank you." He hung up and stared accusingly at Bob and Wally. "That was the computer center. They said the system went down a half hour ago and that whenever they try to bring it back up, the printer just prints 'Let Wally go.' What do you make of THAT, Mr. Walters?"

Now it was Bob's turn to stare at Wally, who was digging in his pocket for a piece of bubble gum. During the entire meeting, Wally had remained quiet, content to daydream. Now he looked up with a smile first at his father and then at Blackwalder.

"Can I go now? I want to get home so I can watch SESAME STREET."

"You can't go anywhere until I get to the bottom of this," Blackwalder bellowed.

Just then, the lights went out. "What?!" Blackwalder fumbled for the phone. The same buzz blared loudly from the receiver. He got up and banged his shins on the half-open file drawer in his desk. "Miss Garcia! Find that fuse!"

Wally jumped up and tugged at his father's sleeve. "I think we'd better go now." They got up and groped their way out of the office. Halfway down the hall, they could still hear Blackwalder shouting for his secretary.

As they drove home, Bob silently brooded about the odd occurences in Blackwalder's office. He wanted to ask Wally if he knew anything about the missing files. But he was afraid to hear the answer. Finally, he said, "I guess Blackwalder will call the police."

Wally answered immediately and confidently. "I don't think so."

"That's what I was afraid you'd say," Bob said, looking over at Wally, who merely smiled.

When they got home, Wally went straight to the computer terminal and started to type. After just a short time, he got up and went into the den and turned on the TV. The hyperactive voices singing the SESAME STREET theme soon drifted into the living room where Bob was. Without taking his coat off, Bob walked over to the terminal and read its output.

*Thanks, guys.
DON'T MENTION IT, BOSS.
DON'T FORGET SESAME STREET.

"I can see that I need to spend more time with my son," Bob said to himself. He began to chuckle.

Have you written Software for your Altair™ Computer?

The Altair 8800 computer was the first micro produced for the general public and remains number one in sales, with more than 8,000 mainframes in the field. The wide acceptance of the Altair computer and its rapid adaptation to many diversified applications has truly turned the dream of the affordable computer into a reality.

Yet the machine itself, remarkable as it is, represents only the beginning. The right Software, tailored to meet a user's specific requirements, is a vital part of any computer system. MITS wants to insure that Altair users everywhere have the best applications software available today and in the future. For this reason, a new MITS subsidiary, the ALTAIR SOFTWARE DISTRIBUTION COMPANY, has been formed. Its purpose: to acquire the highest quality software possible and distribute it nationally through Altair Computer Centers.

That's where you come in. The ASDC will pay substantial royalties to the originators of all software accepted into the ASDC library. If you have written business, industrial or commercial use software for the Altair 8800, ASDC wants to hear from you. It is the aim of the ASDC to stimulate and reward creativity in producing useful software that makes those dreams of "computers for everyone" come true. The ASDC will select only software that measures up to its high standards for system design, coding and documentation. The software will then be further documented and distributed through Altair Computer Centers around the country.

For more information on how to submit software to the ASDC, ask your Local Altair Computer Center for an ASDC Software Submittal Packet or contact the ALTAIR SOFTWARE DISTRIBUTION COMPANY.

A subsidiary of MITS

ASDC

ALTAIR SOFTWARE DISTRIBUTION COMPANY

3330 Peachtree Road, Suite 343 Atlanta, Georgia 30326 404-231-2308